BCS Tools/Nexto - Modbus TCP and Crevis I/O

KI00350 2013-08

1 Function and area of use

This Start Up document describes the necessary steps to be taken when configuring Modbus TCP with Crevis I/O in a Nexto system.

2 About this Start Up document

This Start Up document should not be considered as a complete manual. It is an aid to be able to start up a normal application quickly and easily.

In this document the following software and hardware has been used.

Software

- BCS Tools 1.31

Hardware

- Beijer Electronics NX3010 (CPU)
- Decentralized I/O (Crevis I/O)

This document and other Start Up documents can be obtained from our homepage. Please use the address *manual@beijer.se* for feedback on our Start Up documents.

Beijer Electronics Automation AB – *a company in the Beijer Electronics Group*

Parent Company (Reg. office)
Beijer Electronics Automation AB
P.O. Box 426
SE-201 24 MALMÖ, SWEDEN
Telephone +46 40 35 86 00
Fax +46 40 93 23 01
Visitor's address Stora Varvsgatan 13A, Malmö

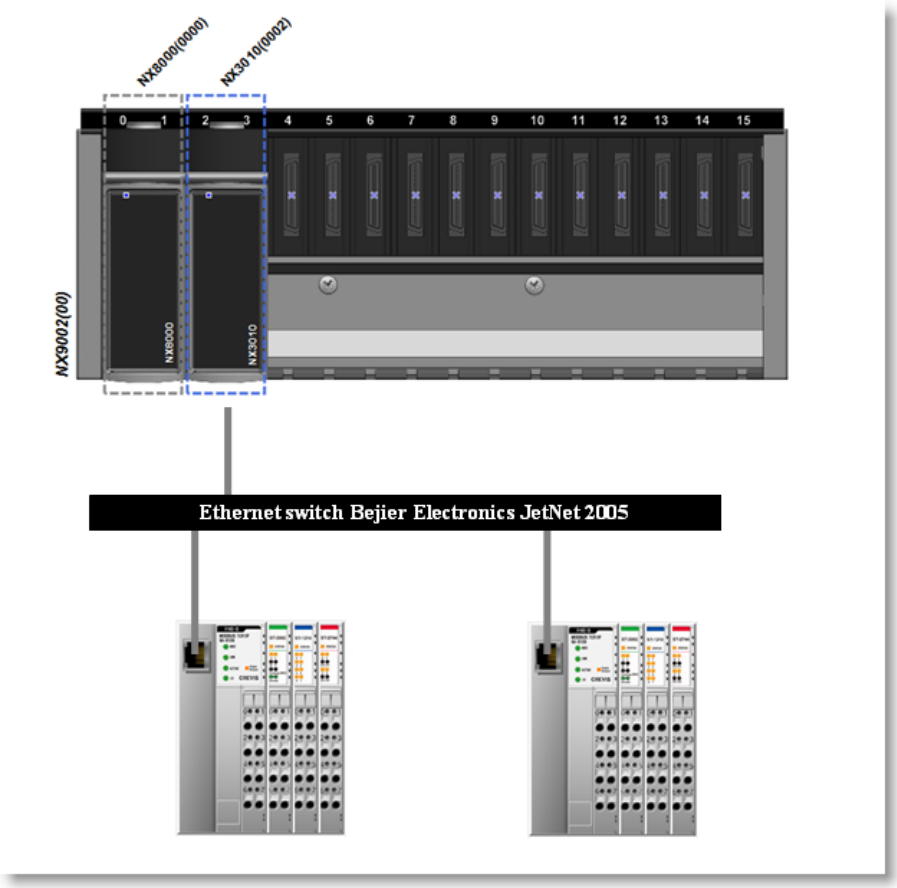
Subsidiaries
Norway, Drammen: Beijer Electronics AS, ❩ +47 32 24 30 00
Finland, Helsinki: Beijer Electronics Oy, ❩ +358 207 46 35 40
Denmark, Copenhagen: Beijer Electronics A/S, ❩ +45 70 26 46 46
Estonia, Tallin: Beijer Electronics Eesti Oü, ❩ +372 6 518140
Latvia, Riga: Beijer Electronics SIA, ❩ +371 7 842280
Lithuania, Vilnius: Beijer Electronics UAB, ❩ +370 5 2323101

Reg no. 556701-3965 VAT no SE556701-3965-01, Internet www.beijer.se, e-mail info@beijer.se

3 Table of Contents

4 Overview

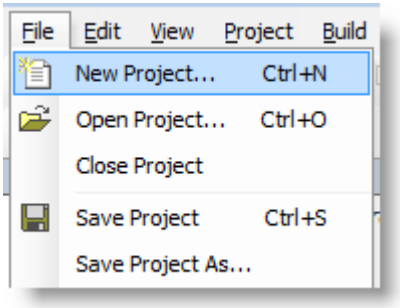
This is the basic configuration. The communication between the NX3010 and the NA-9189 uses the built-in ethernet port of the NX3010.



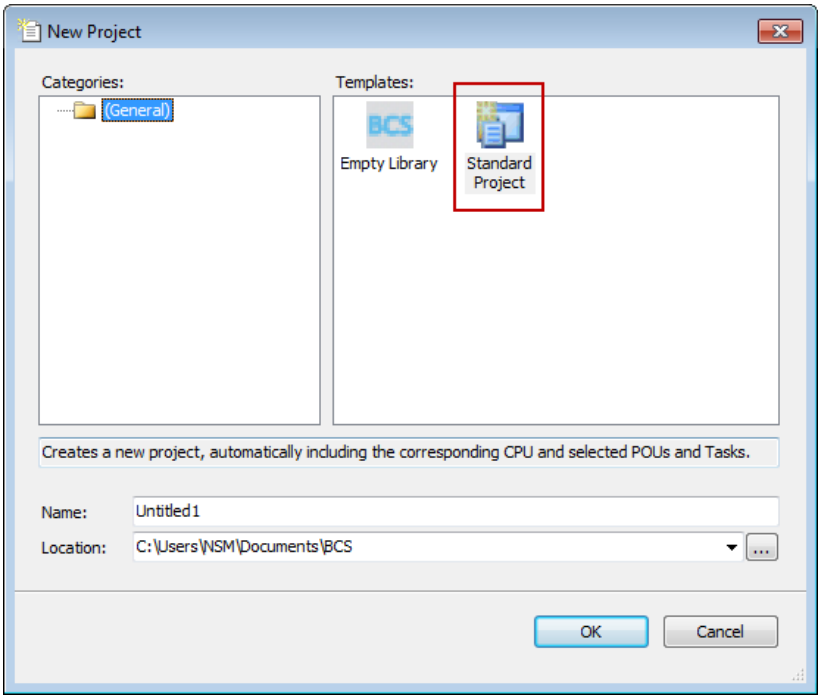
5 Configuration in BCS Tools

5.1 Create a new project

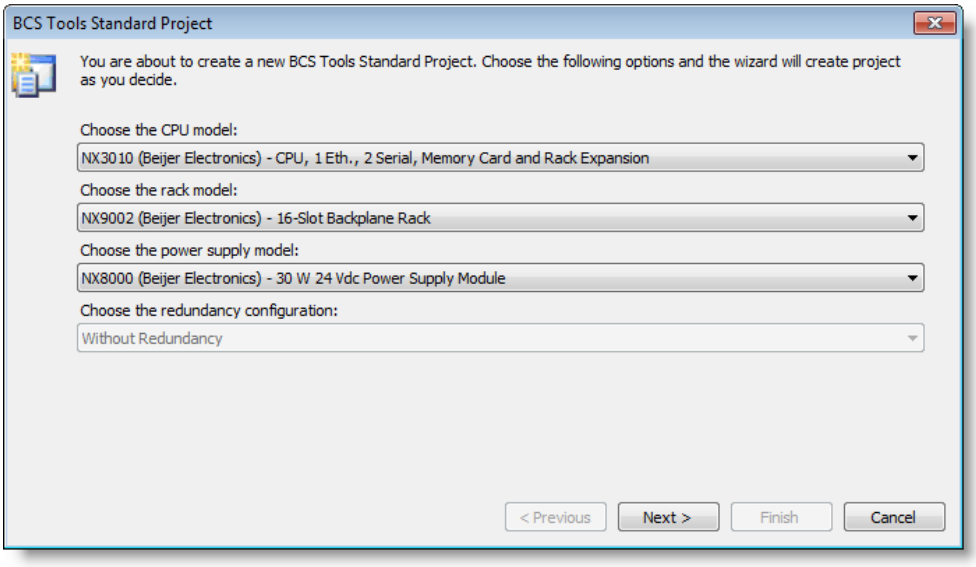
Create a new project from the File menu.



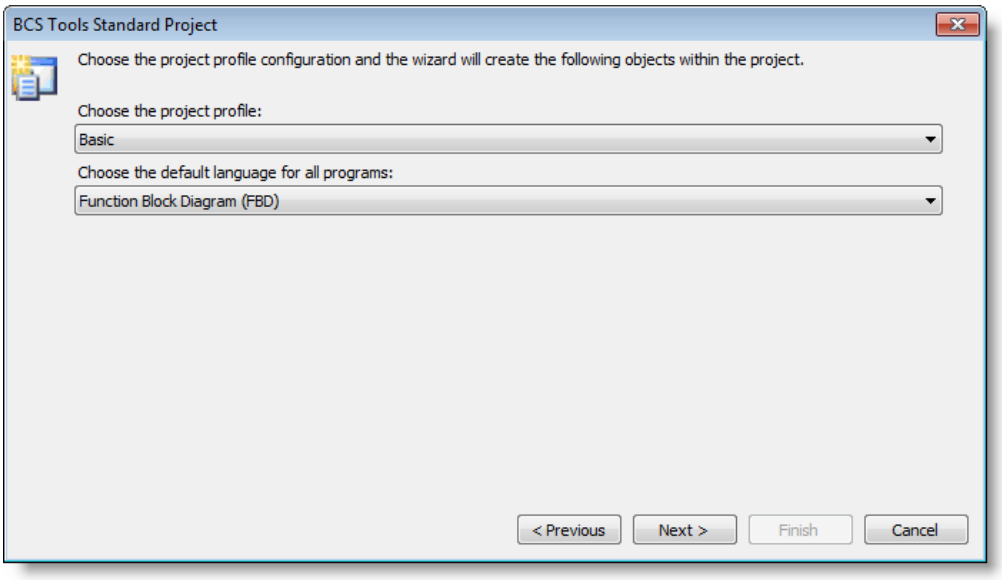
Choose the path of where the project should be created and the standard template.



Choose the cpu, rack and power supply model.



Choose the project profile, in this case the basic is being used.



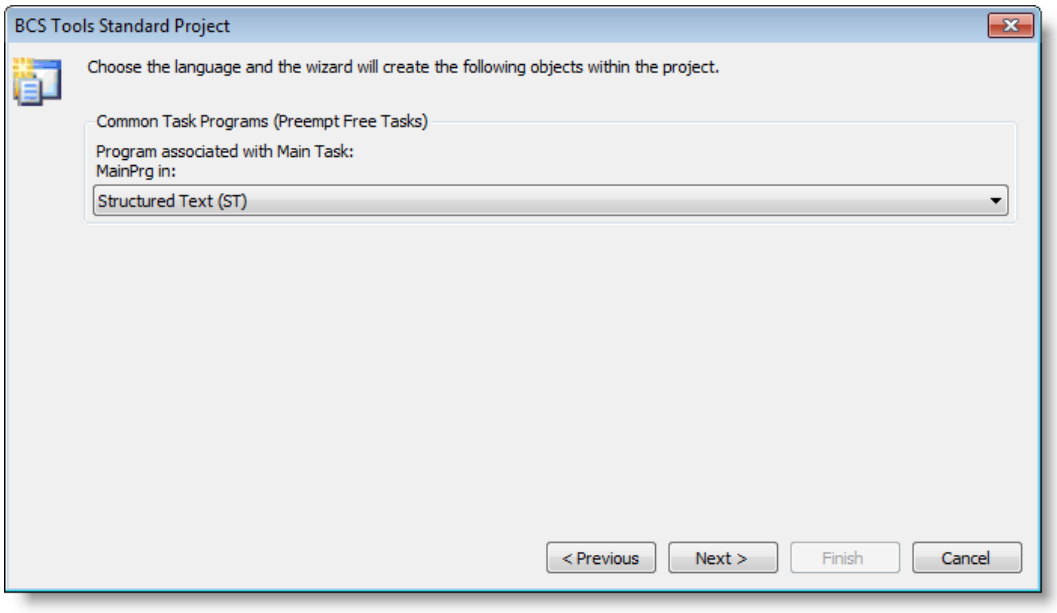
In the Basic project profile, the application has a user task from the Freewheeling type called MainTask which executes the program in a continuous loop (with no definition of cycle time) with priority fixed in 13. This task is responsible for the execution of a single programming unit POU called MainPrg. It is important to stress that the cycle time can vary according with communication task quantity used, as in this mode, the main task is interrupted by communication tasks.

This profile allows the inclusion of two interruption tasks with higher priority which can preempt the MainTask at any moment: the task called ExternInterruptTask00 is an interruption task from the Extern type with priority fixed in 02; the task called TimeInterruptTask00 is an interruption task from the cyclic type with priority fixed in 1.

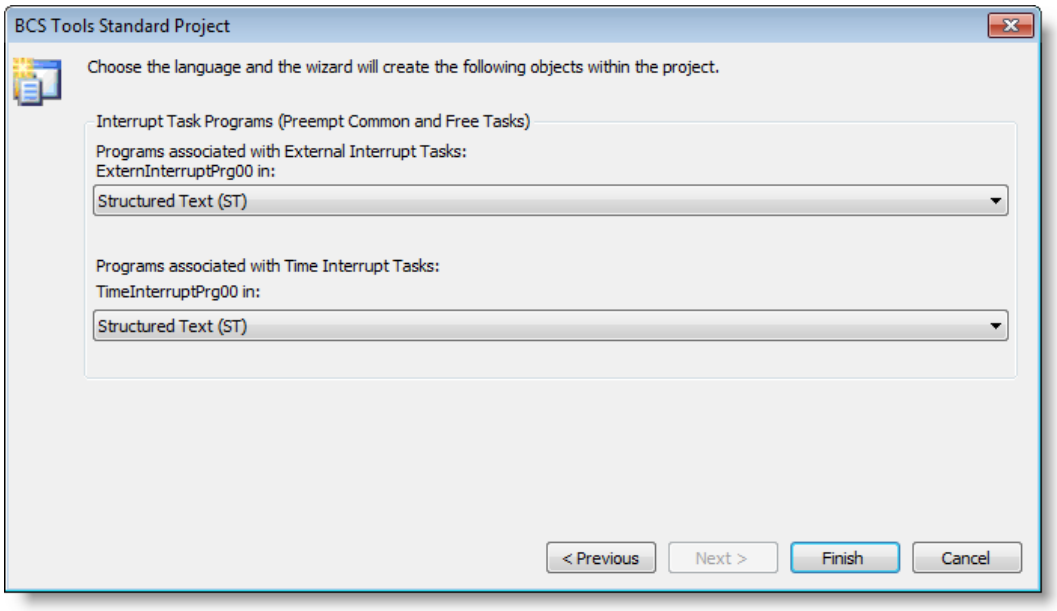
Task	POU	Priority	Type	Interval	Event
MainTask	MainPrg	13	Freewheeling		-
ExternInterruptTask00	ExternInterruptPrg00	02	Extern		IO_INT_0
TimeInterruptTask00	TimeInterruptPrg00	01	Cyclic	20 ms	-

For details about the other profiles refer to the CPU User’s manual.

Choose the prefered programming language for the freewheeling task.



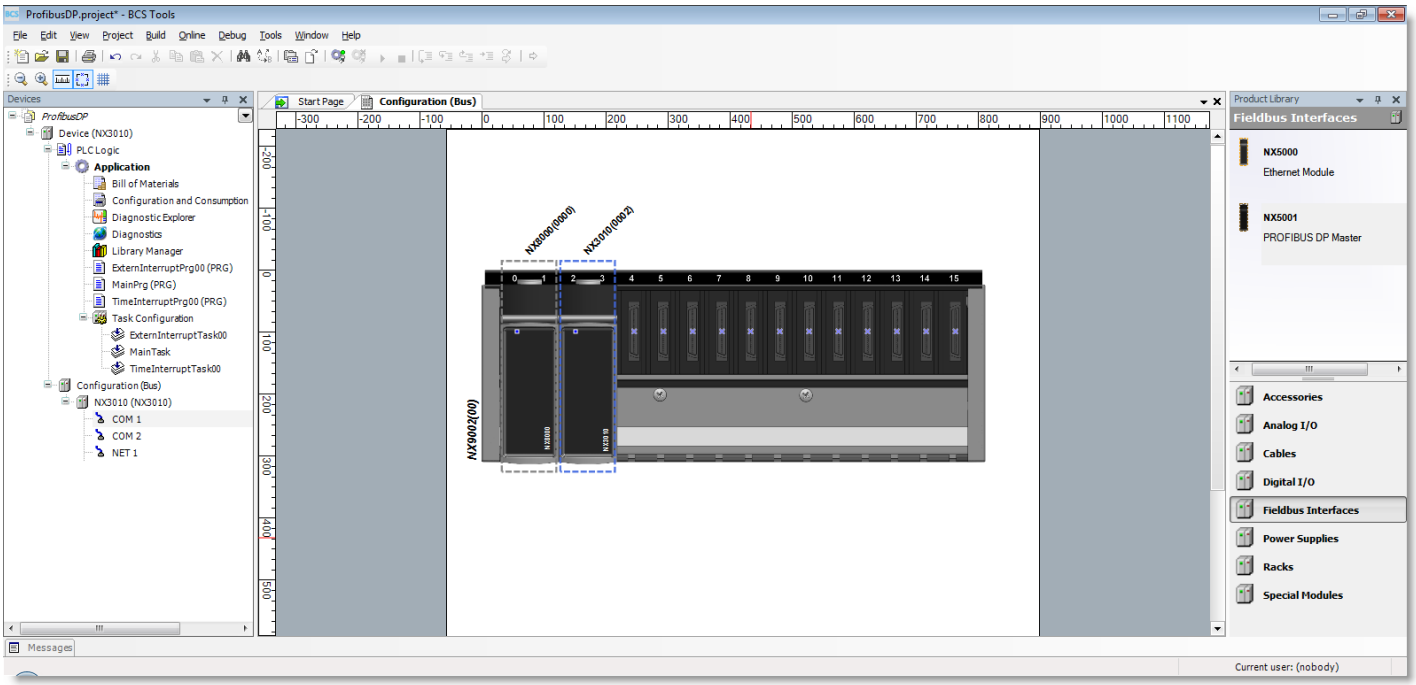
Choose the prefered programming language for the interrupt tasks.



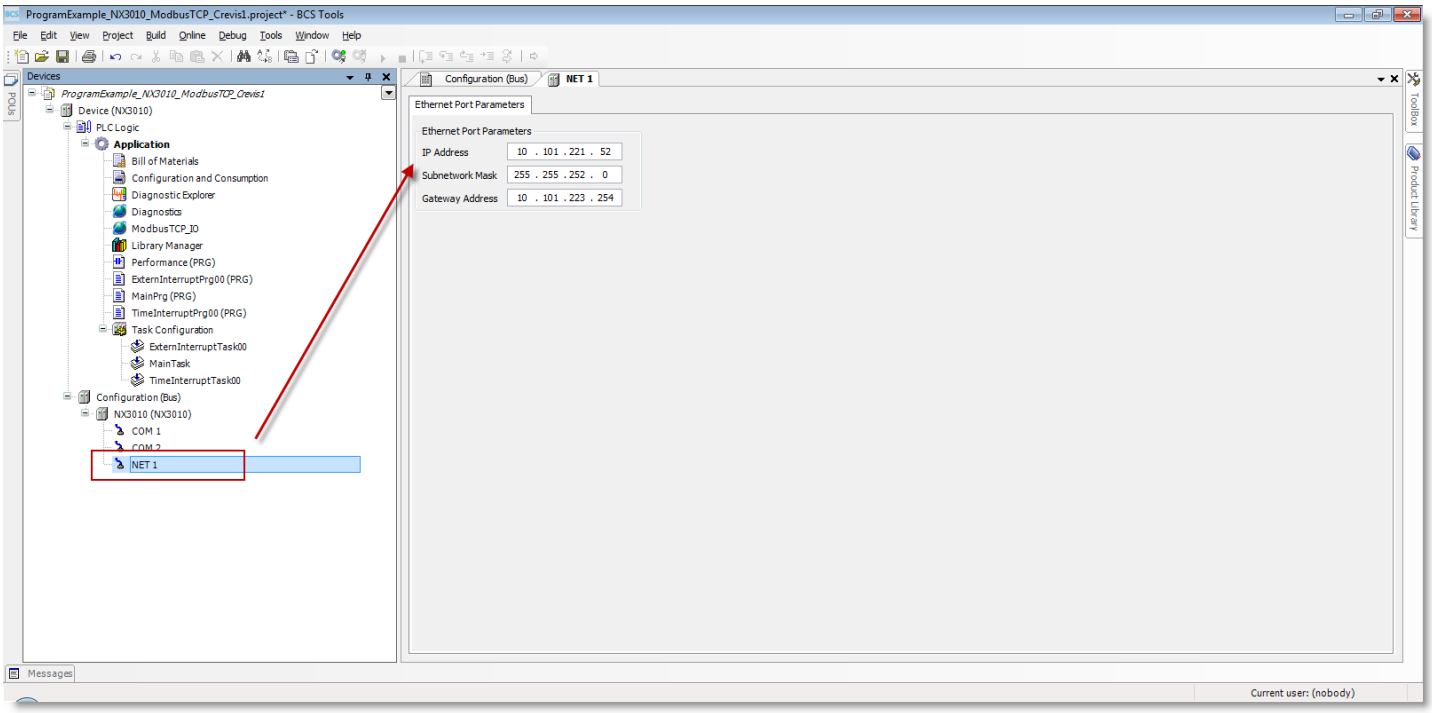
5.2 Configure the IP address of NET1 interface

When the wizard is completed the following view appears, this is where the hardware configuration of the system is made. In this start up we will use the following IP adresses:

NX3010 = 10.101.221.52, **NA9189_1** = 10.101.221.53, **NA9189_2** = 10.101.221.54



Double-click the NET1 icon in the project tree and enter the IP details of the NX3010.



5.3 Configure the IP address of NA-9189

Factory settings

IP-address 192.168.123.1

Subnet mask 255.255.255.0

Default Gateway 192.168.123.254

The addess can be changed in two ways.

- The basic way is by using ARP commands from the command promt. This is explained further down.
- The most intuitive way to change the addresses is to use the "Bootp server tool" in the software "IO Guide Pro", currently **Windows XP** is recommended to use.

To setup using ARP commands follow these steps. Note! The IP can only be changed to another within the same subnet.

Command Prompt

>ping 192.168.123.236 //current IP address

>arp -a //view Ethernet physical address

>arp -d 192.168.123.236 //delete arp table

>arp -s 192.168.123.237 00-14-F7-00-00-00 //assign static arp table with new IP address

//"00-14-F7-00-00-00" is Ethernet Address (See Adapter Label)

>ping -n 1 –l 741 192.168.123.237 //assign new IP address

>arp -d * //clear all arp table

>ping 192.168.123.237 //check response of adapter new IP address

New IP-Address setup.

IP Address = 192.168.123.237

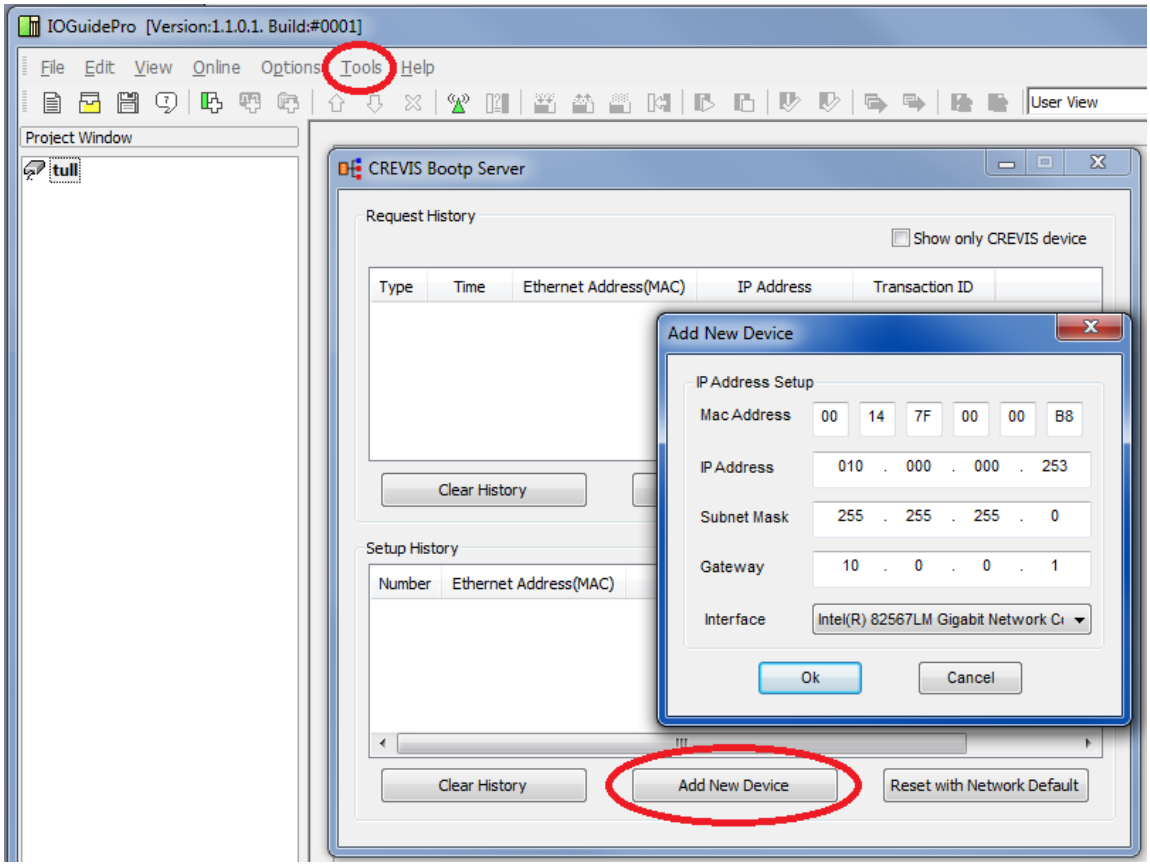
Subnet Mask = 255.255.255.0

Gateway = 192.168.123.254

To setup the IP address using **IO Guide Pro** follow these steps:

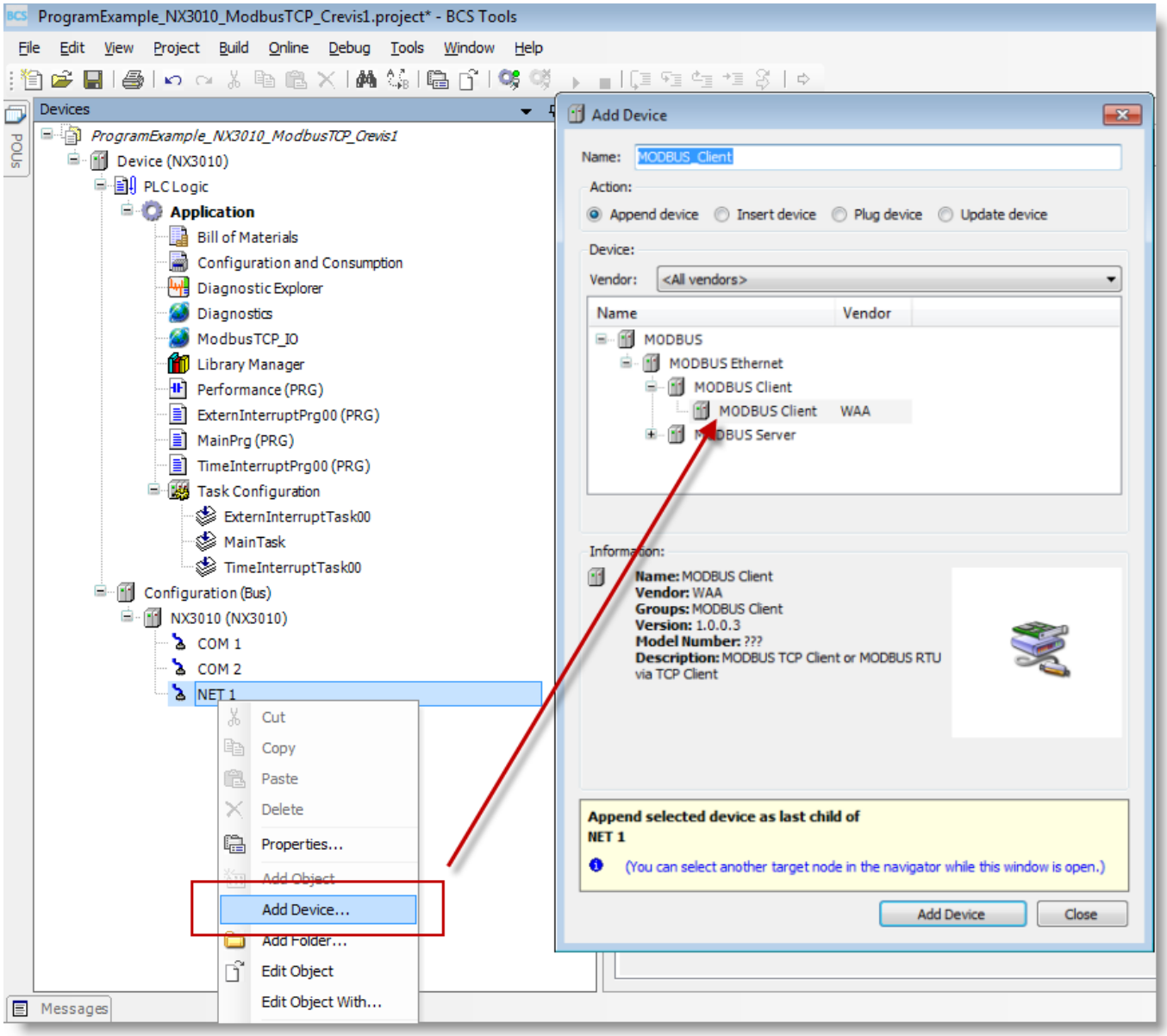
- Start IO Guide Pro
- In the "Tools" menu, select "Bootp server".
- Click the "Add New Device" button at the bottom.
- Enter the requested parameters. (the MAC address is printed on the node module)
- Click Ok, close down the Bootp tool, and try to ping the unit.
- Now you can create a new project in IO Guide Pro using the NA-9189 bus module.

- Run "Autoscan" in IO Guide Pro to receive the configuration. (the green "antenna" icon sown belowe)
- You can now check and monitor the NA-9189 in IO Guide Pro.



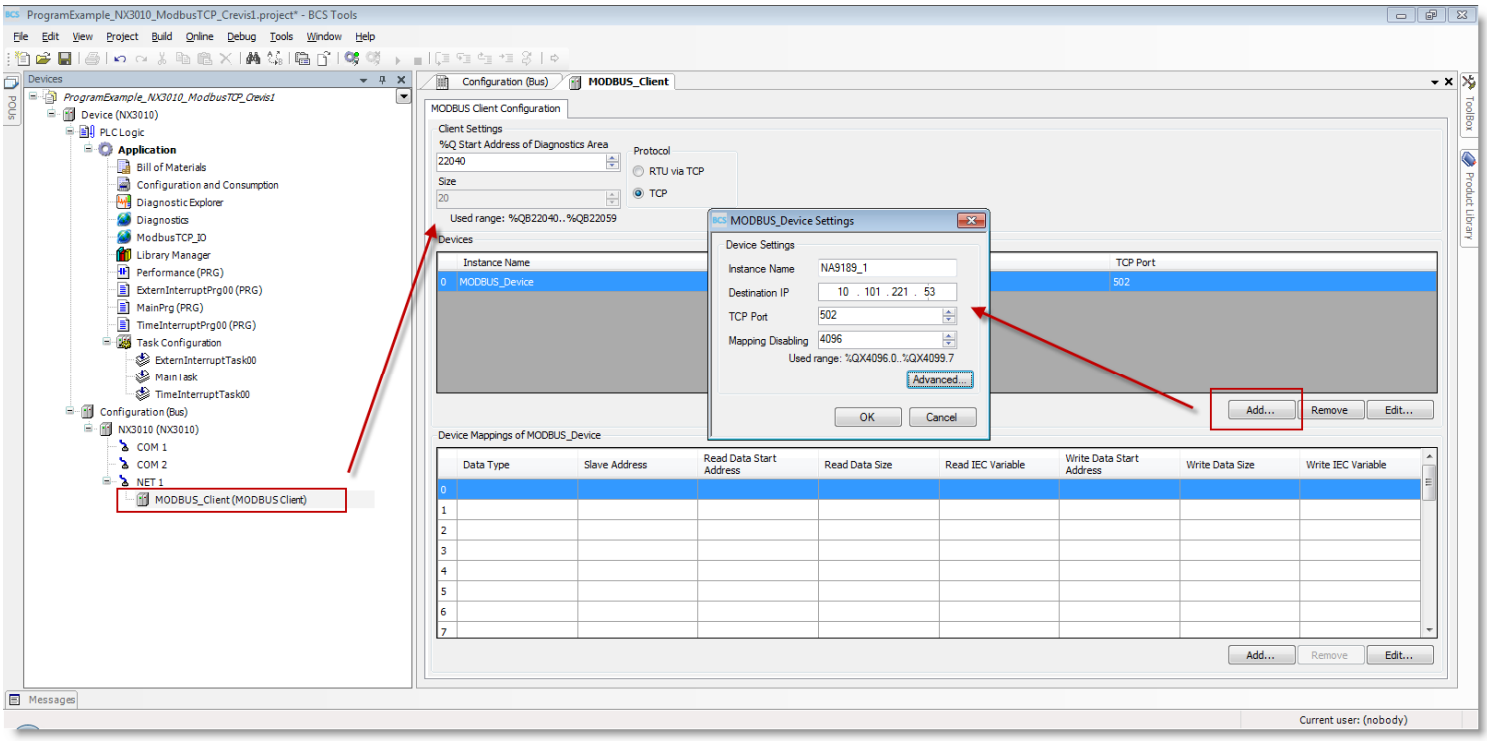
5.4 Add the Modbus Client device

The next step is to insert the Modbus TCP client device on the NET1 interface. This is done by right-clicking the NET1 port and choose Add device and select the Modbus Client device.



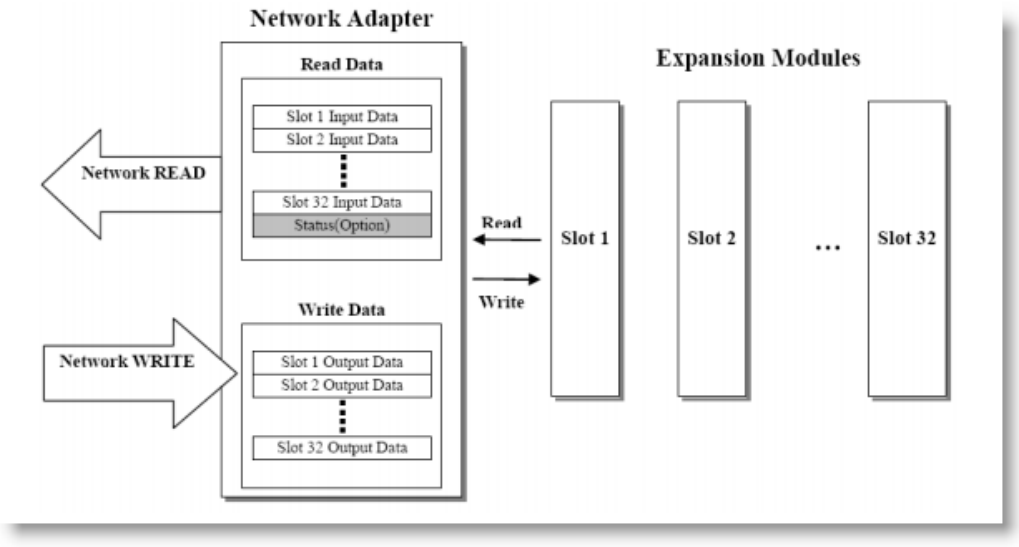
5.5 Add the Modbus TCP slave

To add the Modbus TCP slaves double-click the newly inserted device, Modbus_Client, then click the Add button and enter a name and the IP address of the slave. For further details about other settings refer to the cpu User's manual.



5.6 Data mapping in NA-9189

The NA-9189 module maps all data from/to the I/O modules mounted to one input and one output area.



Input data is mapped as both holding register and coils and the same applies for the output data.

- **Register Map**

Start Address	Read/Write	Description	Func. Code
0x0000 ~	Read	Process input image registers (Real Input Register)	4, 23
0x0800 ~	Read/Write	Process output image registers (Real Output Register)	3, 16, 23
0x1000 ~	Read	Adapter Identification special registers.	3, 4, 23
0x1020 ~	Read/Write	Adapter Watchdog, other time special register.	3, 4, 6, 16, 23
0x1100 ~	Read/Write	Adapter Information special registers.	3, 4, 6, 16, 23
0x2000 ~	Read/Write	Expansion Slot Information special registers.	3, 4, 6, 16, 23

* The special register map must be accessed by read/write of every each address (one address).

- **Bit Map**

Start Address	Read/Write	Description	Func. Code
0x0000 ~	Read	Process input image bits All input registers area is addressable by bit address. Size of input image bit is size of input image register * 16.	2
0x0800 ~	Read/Write	Process output image bits All output registers area is addressable by bit address. Size of output image bit is size of output image register * 16.	1, 5, 15

The two NA-9189 used in this example are equipped with the following modules.

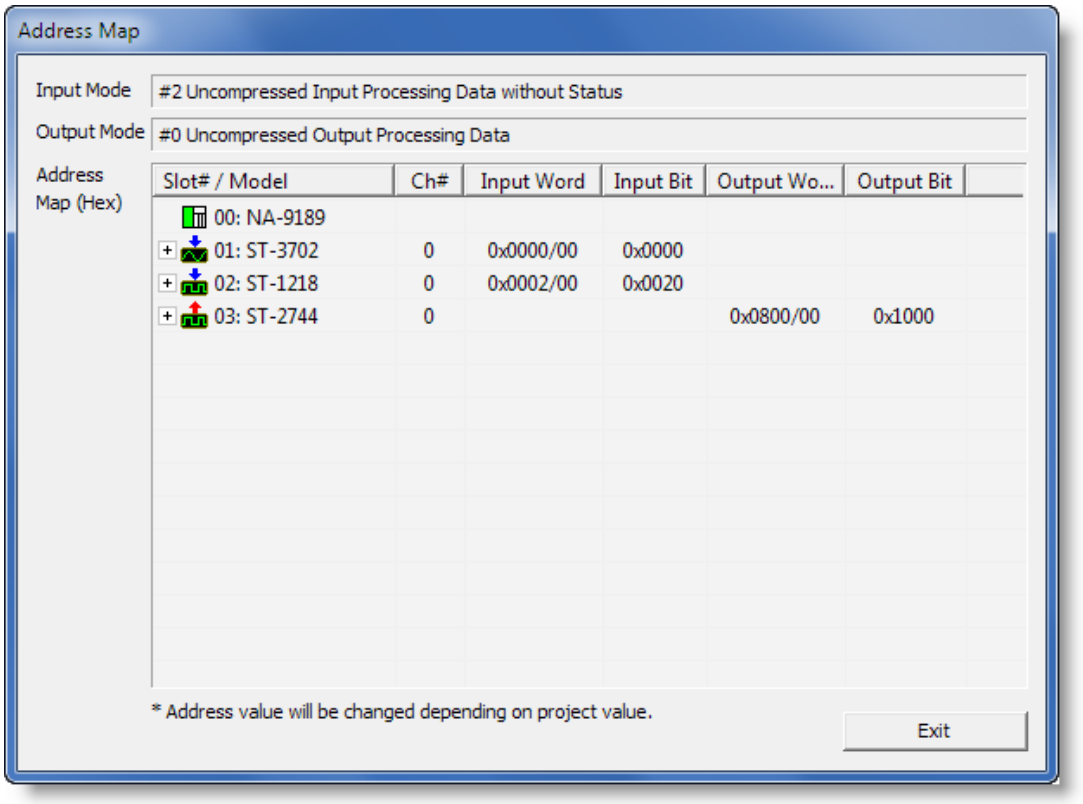
NA-9189 | ST-3702 | ST-1218 | ST-2744

ST-3702 = 2 RTD inputs

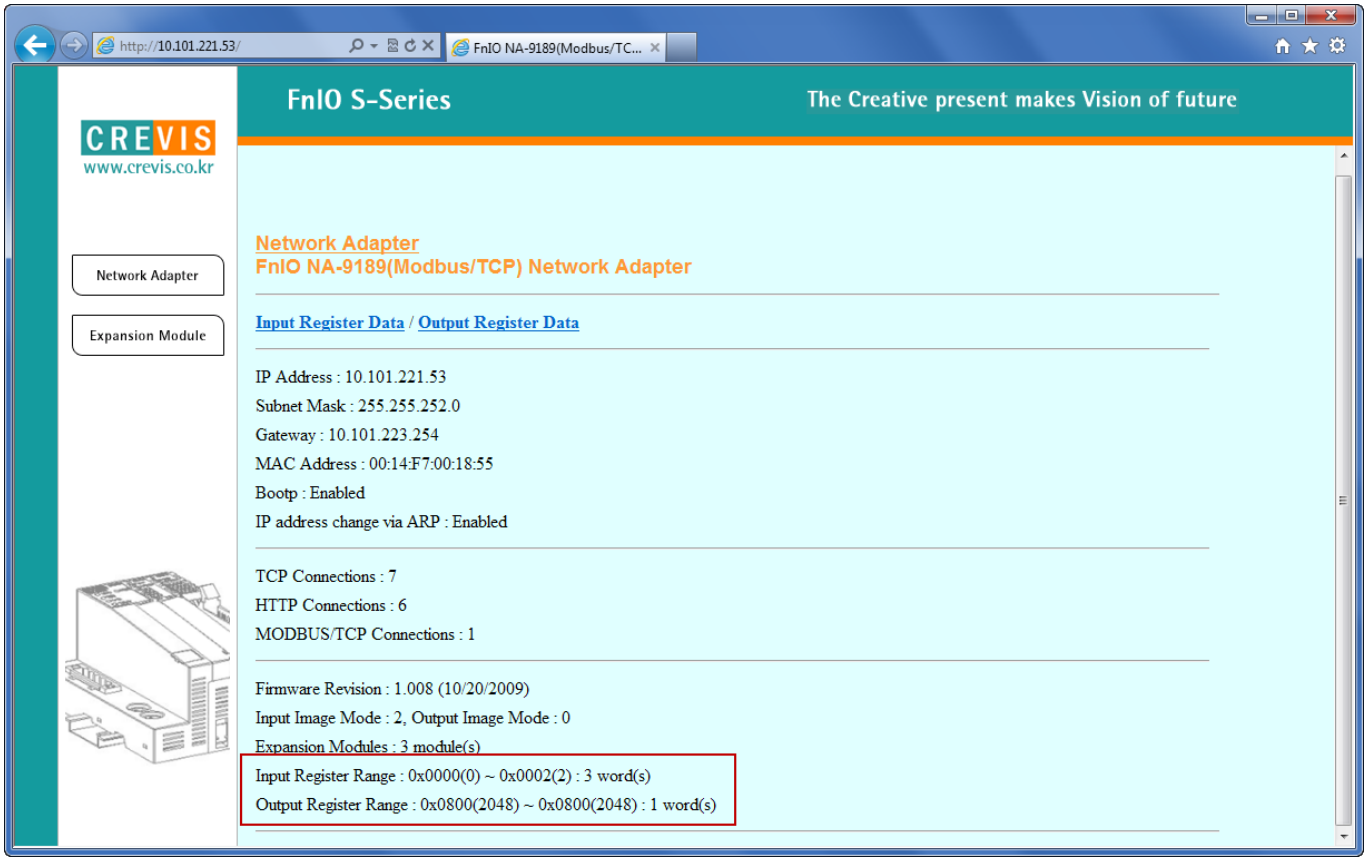
ST-1218 = 8 DI

ST-2744 = 4 DO

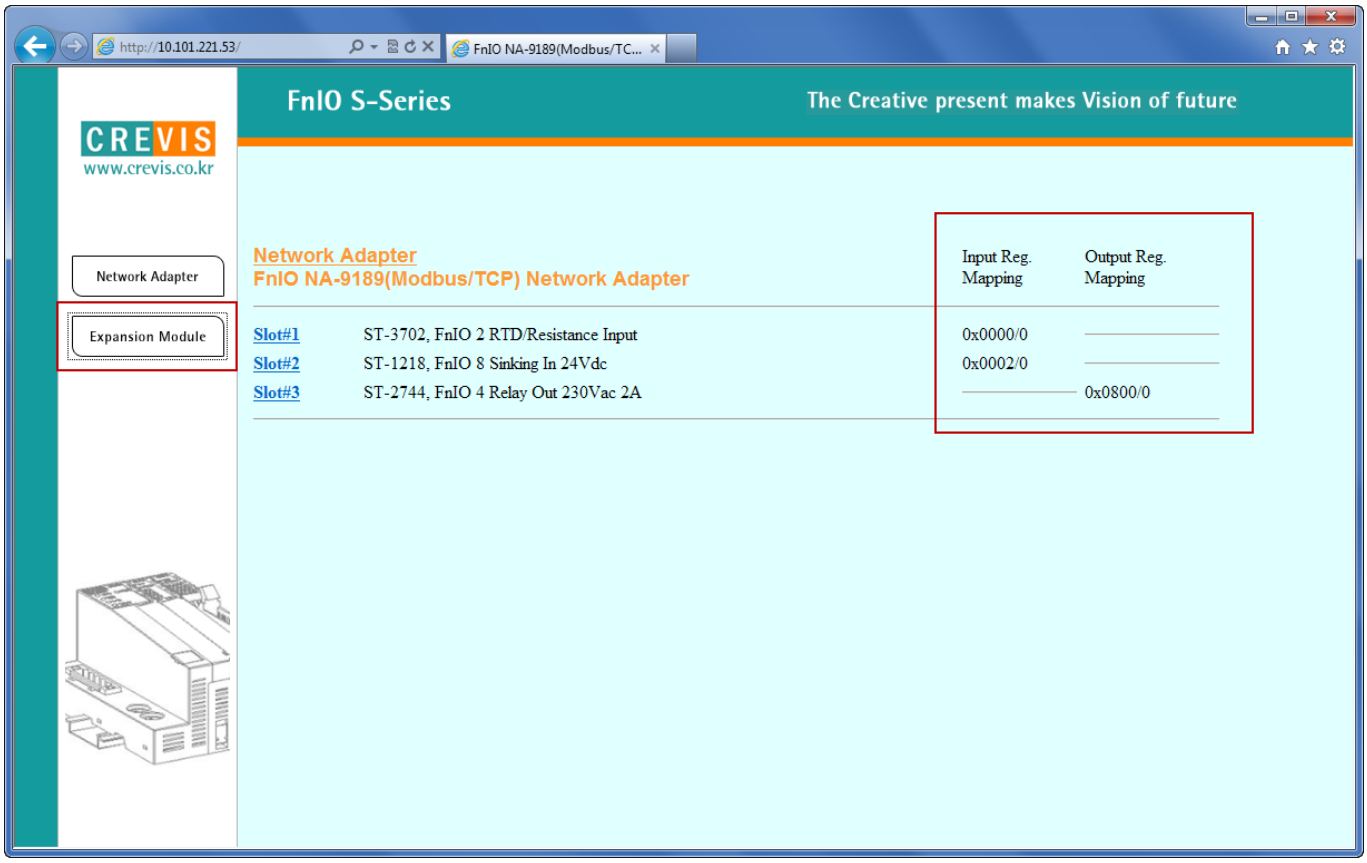
The address mapping can be checked by using IO Guide Pro from the View menu and select the View address map option.



Another way to check the mapping is to use a web browser and connect to the NA-9189.



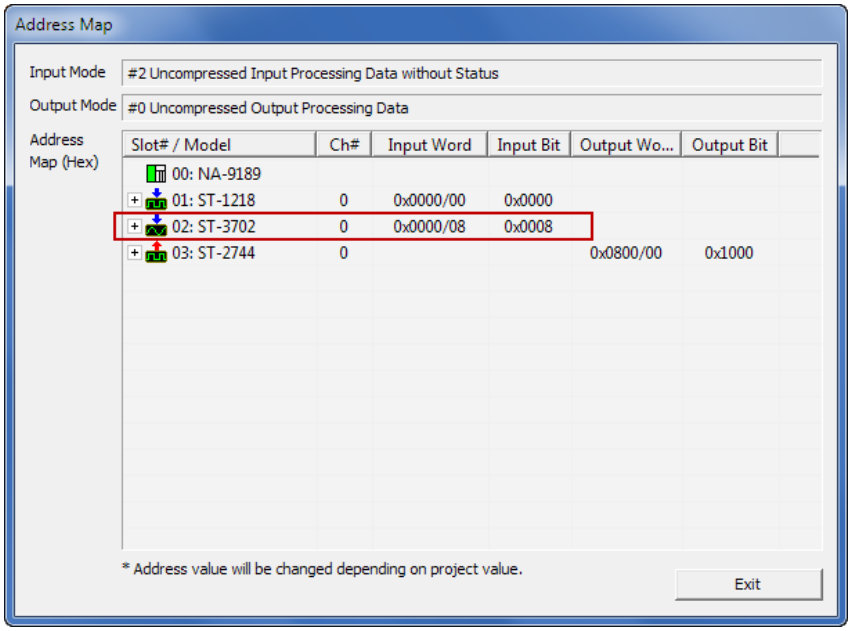
Click the Expansion Module button to get more details of the mapping.



It's recommended to always install analog modules before digital to avoid a possible byte separation of a 16bit input word.

The following is a example of a byte separation:

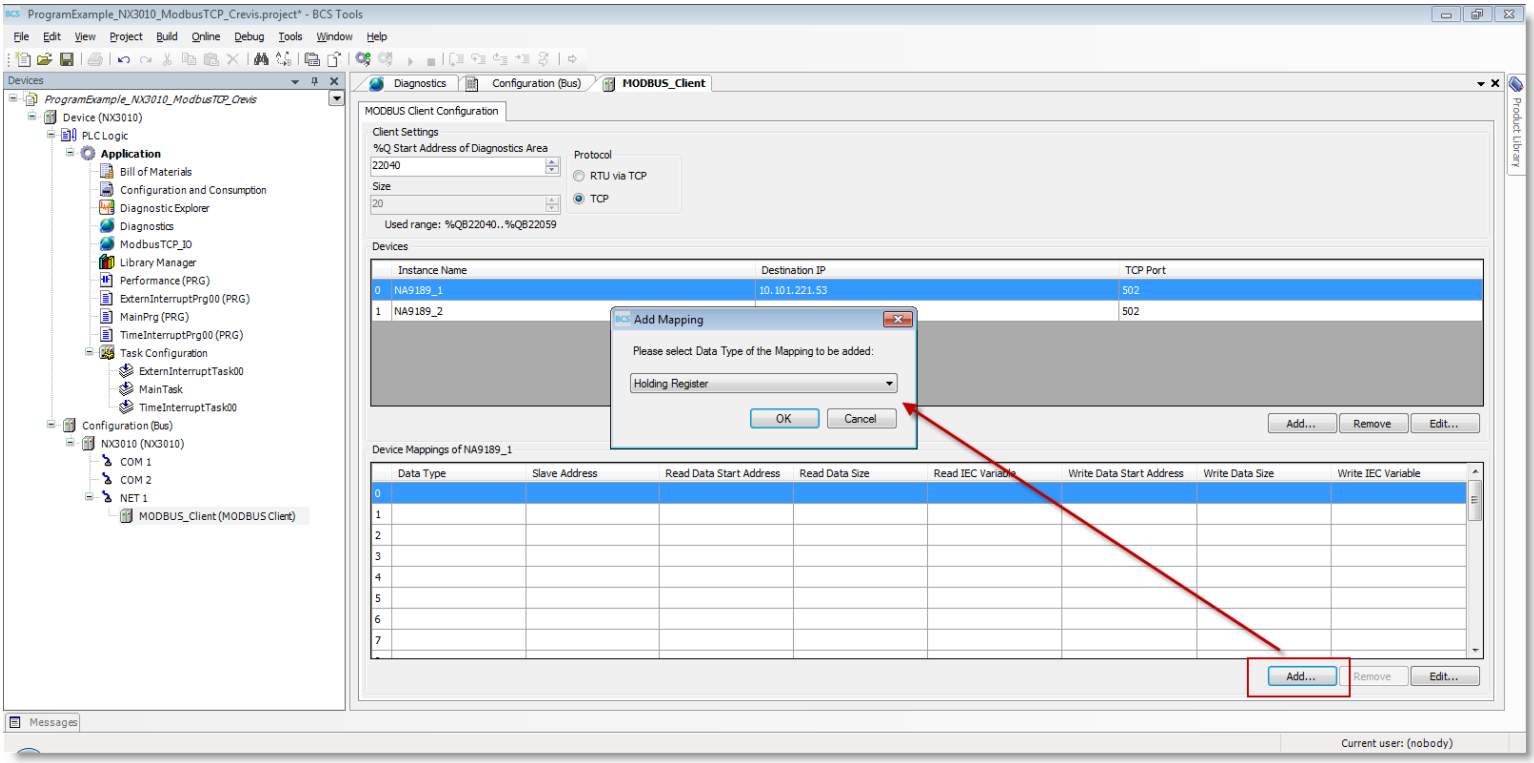
NA-9189 | ST-1218 | ST-3702 | ST-2744



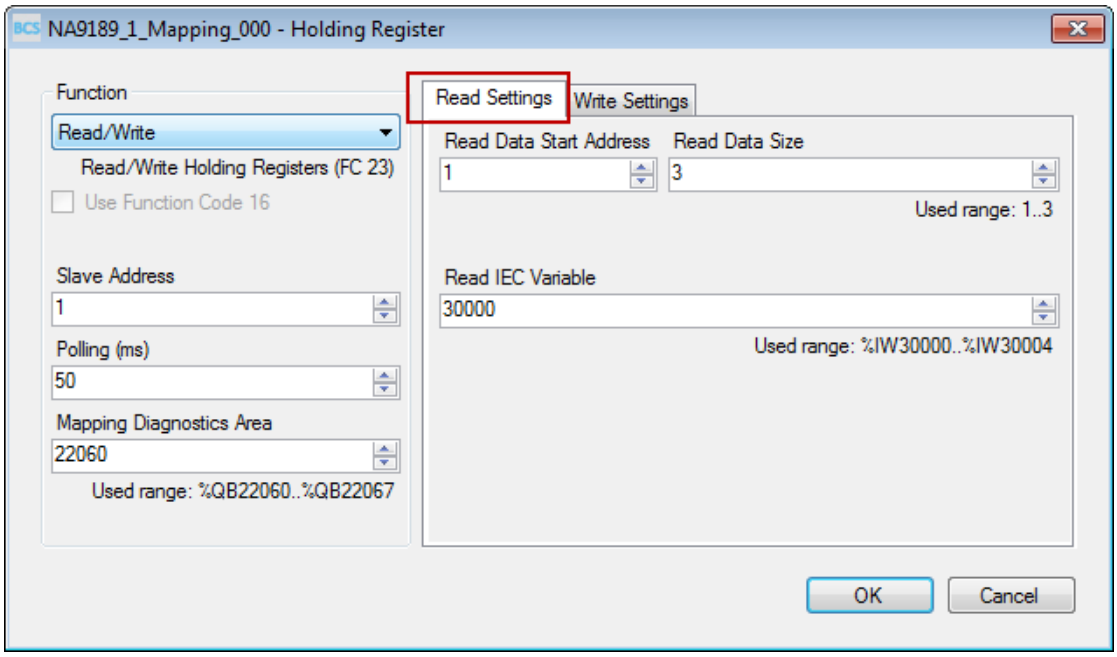
5.7 Configure the data mapping of the Modbus TCP slave

Click the Add button to configure how to access the NA-9189. The most efficient way is to use the Modbus function Read/Write Holding register, therefore Holding register is chosen.

Click OK to verify the setting.



The next step is to configure how many holding register to read and write and from what address. The NA-9189 input area starts at address 0 and the output area starts at address 2048. The NA-9189 has 0-based addressing which means that the lowest address is 0, while the Nexto system is 1-based which means that the lowest address is 1. This will give us a offset with 1 when addressing a NA-9189.



Function = Read/Write

Slave address = Not used when using NA-9189, use default value

Polling = Set the desired polling rate

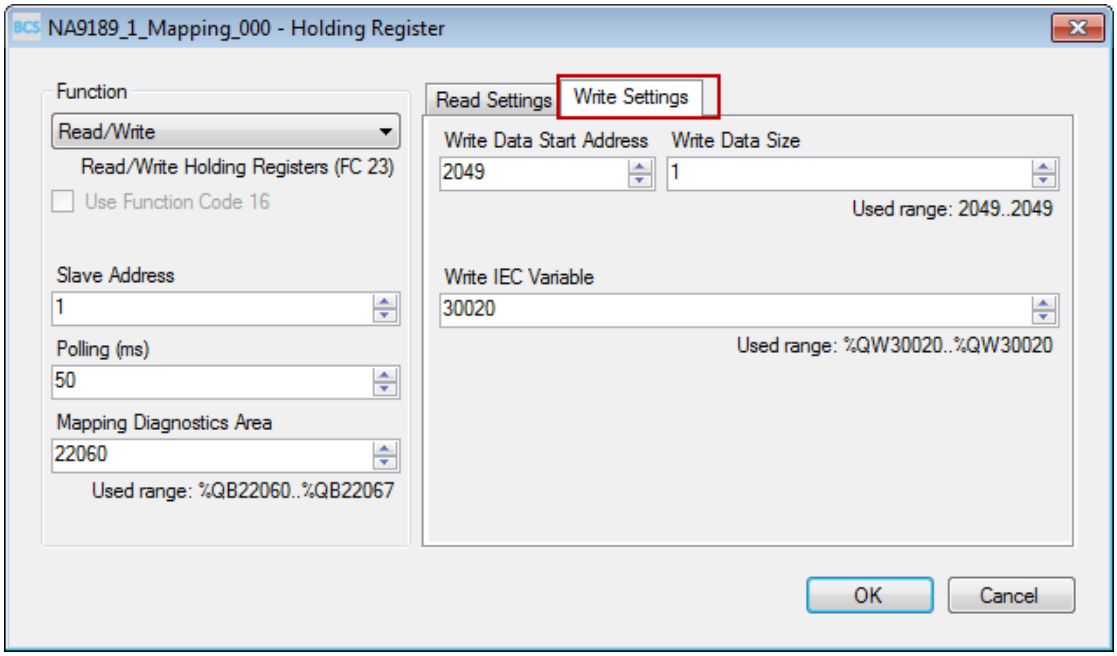
Mapping Diagnostics Area = This is the IEC adresses that will contain diagnostics about the mapping function, refer to Cpu User's manual for more details.

Read Settings

Read Data Start Address = Starting address to read from NA-9189 (because of the address offset address 1 will be address 0 in NA-9189)

Read Data Size = Number of Holding registers to read.

Read IEC Variable = The IEC address to map the read Holding registers.



Write Settings

Write Data Start Address = Starting address to write in NA-9189 (because of the address offset address 2049 will be address 2048 in NA-9189)

Write Data Size = Number of Holding registers to write.

Write IEC Variable = The IEC address to map the written Holding registers.

Repeat the mapping for the other NA-9189. The IEC addresses used for the other NA-9189 is:

Read IEC Variable = 30040

Write IEC Variable = 30060

The first NA-9189 only occupies from from 30000 – 30004 and 3020 – 3020 by creating some "space" between the addresses future expansion will be easier.

The mapping is now complete.

5.8 Create variables and connect to the IEC addresses

To get the Modbus data in user variables the variable needs to be connected to the IEC address. In this example a new GVL is created named ModbusTCP_IO.

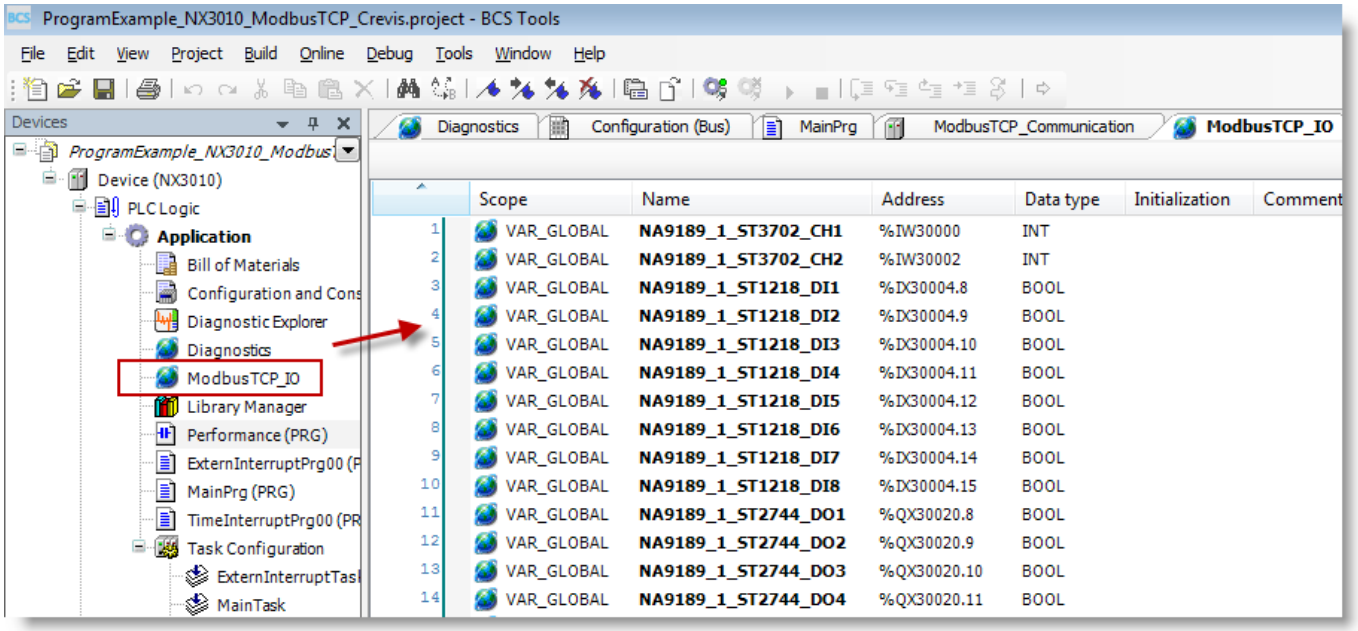
The relation between the Modbus adress in the NA-9189 to the user variable is according the following table. This is the relation for the **NA-9189_1**

	A	B	C
1	**IEC address input data**	**NA-9189_1 Modbus Holding register address input data**	
2			
3	%IW30000	0	AI1 ST-3702
4	%IW30002	1	AI2 ST-3702
5	%IX30004.8	2.0	DI1 ST-1218
6	%IX30004.9	2.1	DI2 ST-1218
7	%IX30004.10	2.2	DI3 ST-1218
8	%IX30004.11	2.3	DI4 ST-1218
9	%IX30004.12	2.4	DI5 ST-1218
10	%IX30004.13	2.5	DI6 ST-1218
11	%IX30004.14	2.6	DI7 ST-1218
12	%IX30004.15	2.7	DI8 ST-1218
13	%IX30004.0	2.8	Not occupied
14	%IX30004.1	2.9	Not occupied
15	%IX30004.2	2.10	Not occupied
16	%IX30004.3	2.11	Not occupied
17	%IX30004.4	2.12	Not occupied
18	%IX30004.5	2.13	Not occupied
19	%IX30004.6	2.14	Not occupied
20	%IX30004.7	2.15	Not occupied
21	**IEC address output data**	**NA-9189_1 Modbus Holding register address output data**	
22	%QX30020.8	2048.0	DO1 ST-2744
23	%QX30020.9	2048.1	DO2 ST-2744
24	%QX30020.10	2048.2	DO3 ST-2744
25	%QX30020.11	2048.3	DO4 ST-2744
26	%QX30020.12	2048.4	Occupied by ST-2744
27	%QX30020.13	2048.5	Occupied by ST-2744
28	%QX30020.14	2048.6	Occupied by ST-2744
29	%QX30020.15	2048.7	Occupied by ST-2744
30	%QX30020.0	2048.8	Not occupied
31	%QX30020.1	2048.9	Not occupied
32	%QX30020.2	2048.10	Not occupied
33	%QX30020.3	2048.11	Not occupied
34	%QX30020.4	2048.12	Not occupied
35	%QX30020.5	2048.13	Not occupied
36	%QX30020.6	2048.14	Not occupied
37	%QX30020.7	2048.15	Not occupied

This is the relation for **NA9189_2.**

	A	B	C
39	**IEC address input data**	**NA-9189_2 Modbus Holding register address input data**	
40			
41	%IW30040	0	AI1 ST-3702
42	%IW30042	1	AI2 ST-3702
43	%IX30044.8	2.0	DI1 ST-1218
44	%IX30044.9	2.1	DI2 ST-1218
45	%IX30044.10	2.2	DI3 ST-1218
46	%IX30044.11	2.3	DI4 ST-1218
47	%IX30044.12	2.4	DI5 ST-1218
48	%IX30044.13	2.5	DI6 ST-1218
49	%IX30044.14	2.6	DI7 ST-1218
50	%IX30044.15	2.7	DI8 ST-1218
51	%IX30044.0	2.8	Not occupied
52	%IX30044.1	2.9	Not occupied
53	%IX30044.2	2.10	Not occupied
54	%IX30044.3	2.11	Not occupied
55	%IX30044.4	2.12	Not occupied
56	%IX30044.5	2.13	Not occupied
57	%IX30044.6	2.14	Not occupied
58	%IX30044.7	2.15	Not occupied
59			
60	**IEC address output data**	**NA-9189_2 Modbus Holding register address output data**	
61	%QX30060.8	2048.0	DO1 ST-2744
62	%QX30060.9	2048.1	DO2 ST-2744
63	%QX30060.10	2048.2	DO3 ST-2744
64	%QX30060.11	2048.3	DO4 ST-2744
65	%QX30060.12	2048.4	Occupied by ST-2744
66	%QX30060.13	2048.5	Occupied by ST-2744
67	%QX30060.14	2048.6	Occupied by ST-2744
68	%QX30060.15	2048.7	Occupied by ST-2744
69	%QX30060.0	2048.8	Not occupied
70	%QX30060.1	2048.9	Not occupied
71	%QX30060.2	2048.10	Not occupied
72	%QX30060.3	2048.11	Not occupied
73	%QX30060.4	2048.12	Not occupied
74	%QX30060.5	2048.13	Not occupied
75	%QX30060.6	2048.14	Not occupied
76	%QX30060.7	2048.15	Not occupied

Screenshot of the created varibles connected to the addresses above.

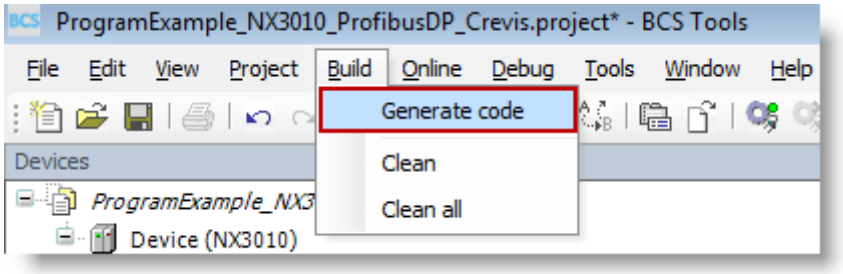


Compile and download of the project

This chapter describes how to compile and download the project.

6.1 Compile of the project

When the configuration has been completed the project has to be compiled. To compile the project click the menu Build\Generate code.



Once the code generation has been successful it's time to download the project.

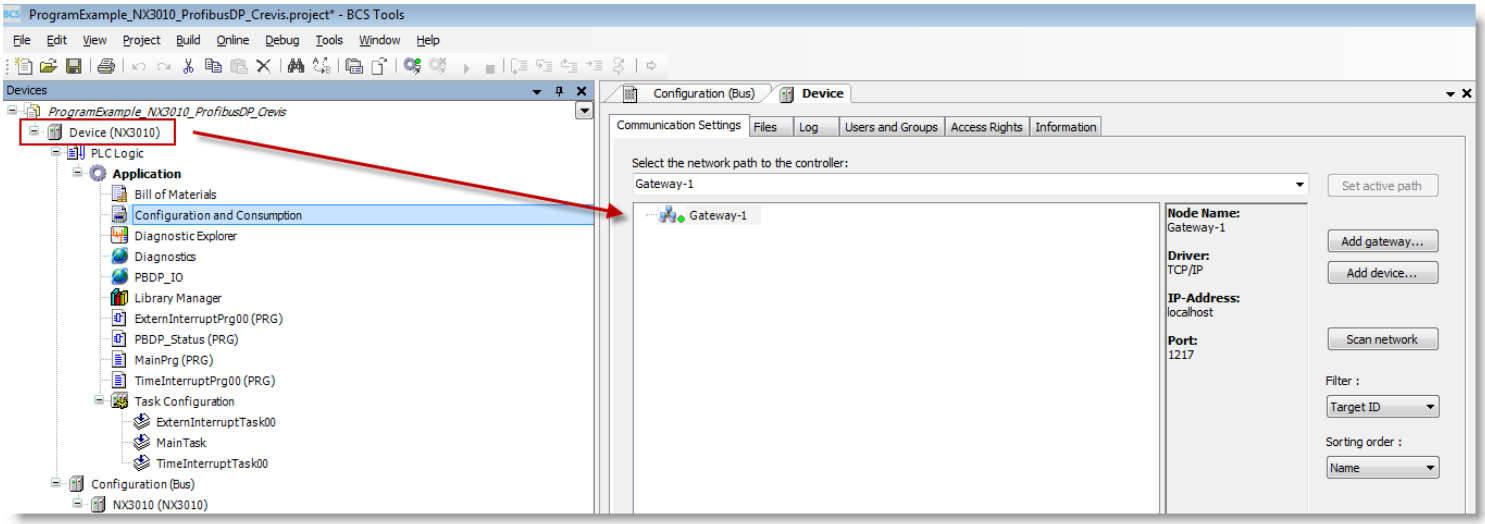
6.2 Download of the project

To download the project the accesspath to the cpu must be configured. The access must be made via the built-in ethernet port of the cpu. The default IP-adress of NET1 is 192.168.15.1

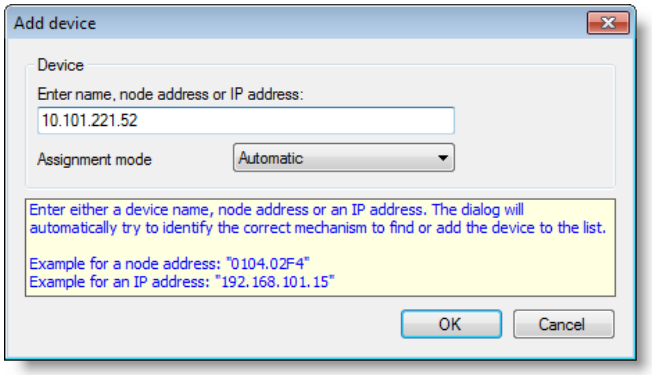
	NET 1
IP Address	192.168.15.1
Subnet Mask	255.255.255.0
Gateway Address	192.168.15.253

In case of using the NX3020 or NX3030 with two ethernet ports NET1 must be used for communication with BCS Tools.

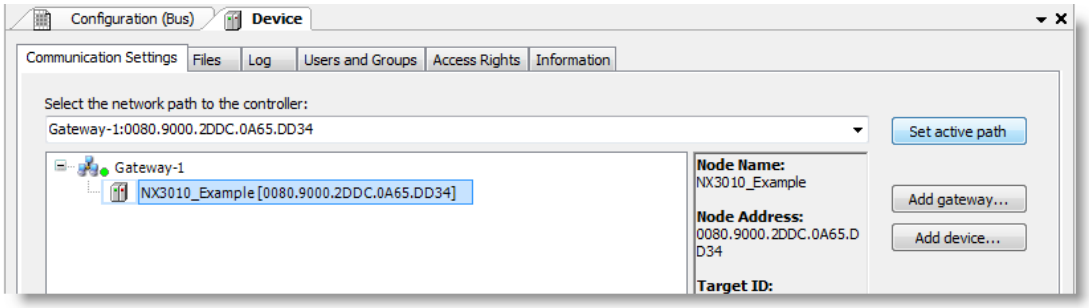
Double-click the device in the project tree to access the communication settings.



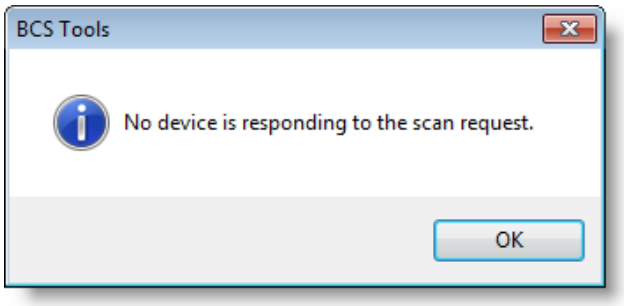
Choose the Scan Network option or add the device manually by clicking the Add device button.



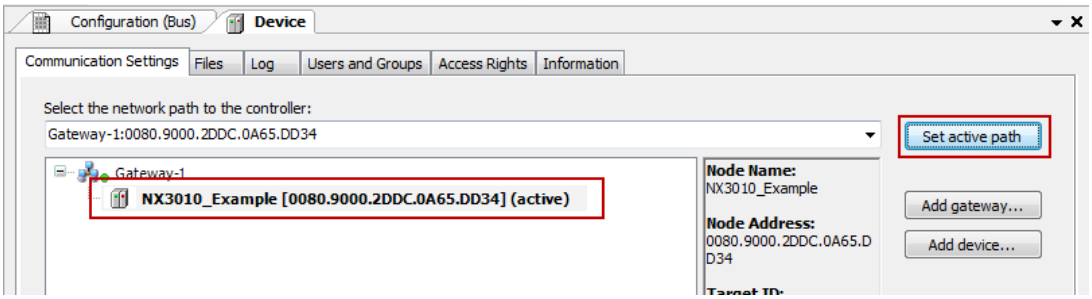
If the cpu is found on the network it will be shown in the list.



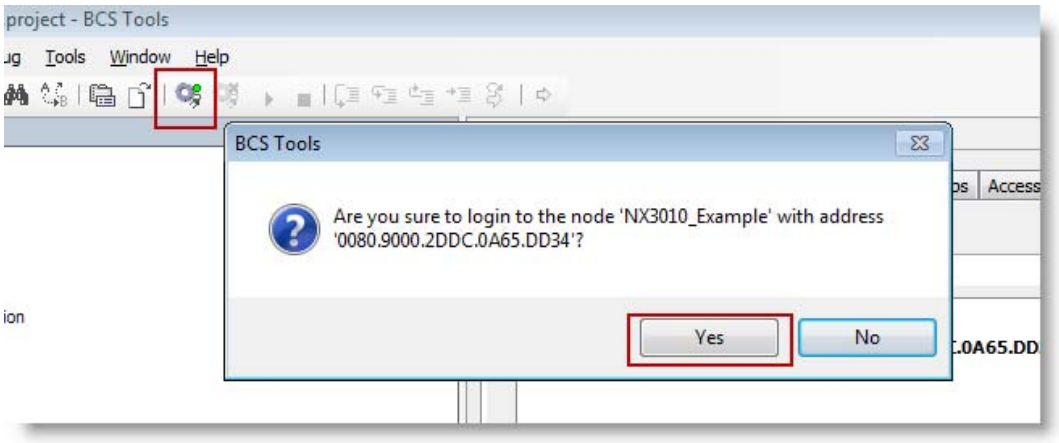
If cpu is unavailable an error message will be shown.



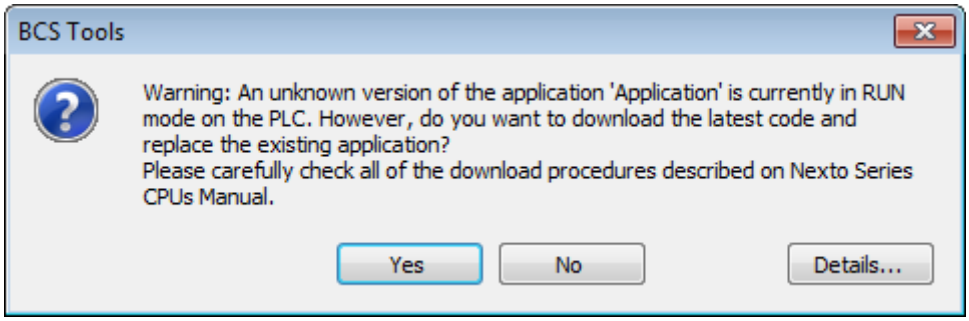
When the cpu has been found click the button Set active path. The cpu in the list will be bold to indicate that this is the path that will be used when an online action is made.



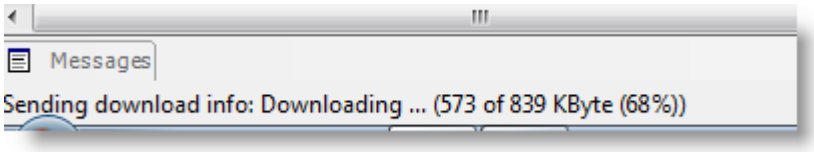
To download the project click the Login button and click the Yes button.



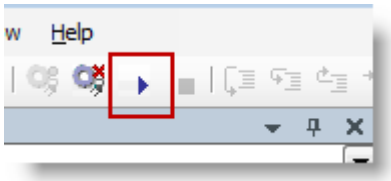
If the cpu is empty or if there is another project running BCS Tools will prompt the user to confirm the download. The cpu will be stopped and the older project will be overwritten.



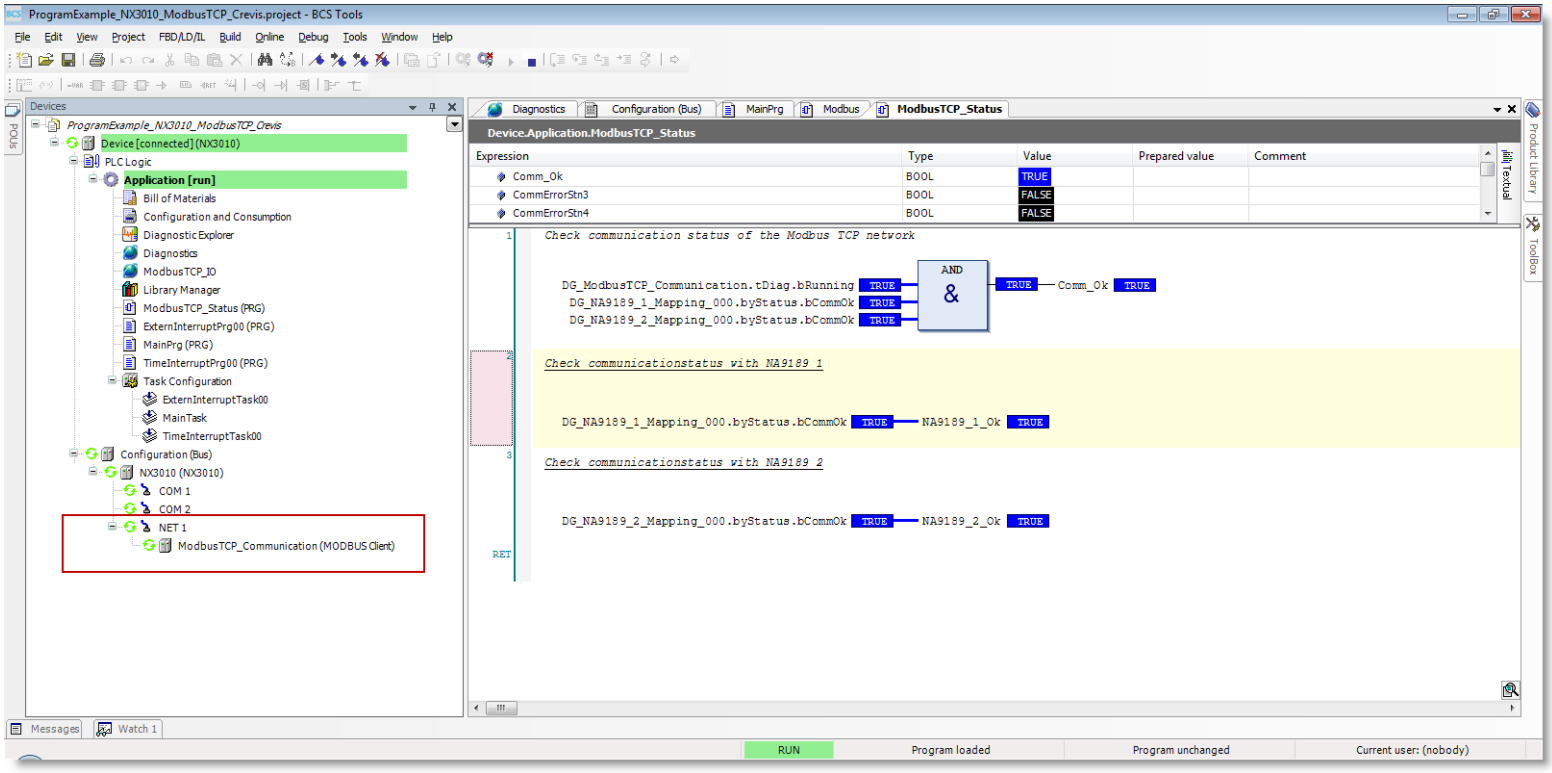
The progress of the download can be found down in the left corner of BCS Tools.



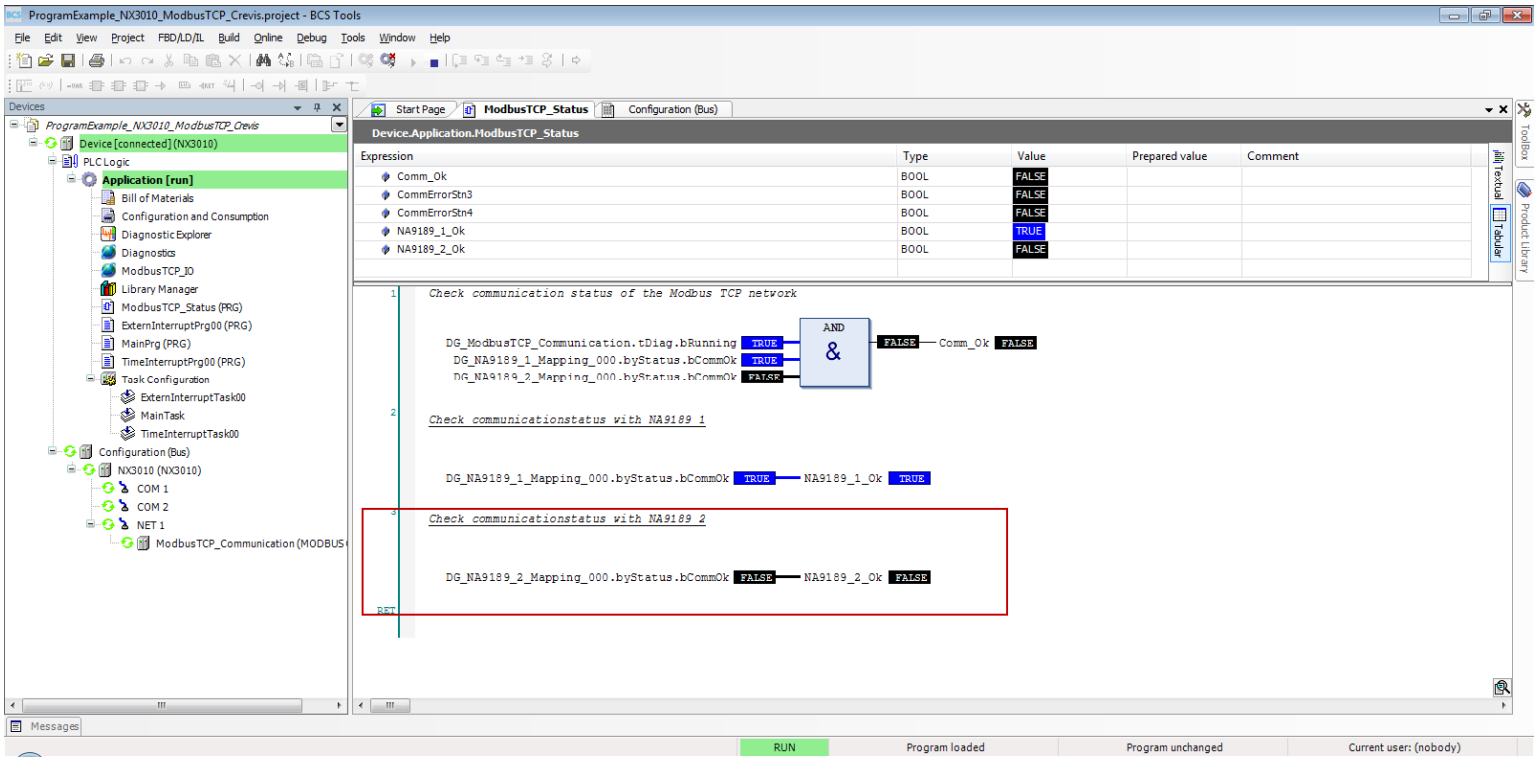
When the download is completed the cpu is in STOP state. To set the cpu into RUN state press the run button. No reset or power on/off is necessary after the download even if parameters has been changed.



The status of the cpu and the Modbus TCP driver can be found in BCS Tools. To see status of each slave the diagnostic variables must be used.



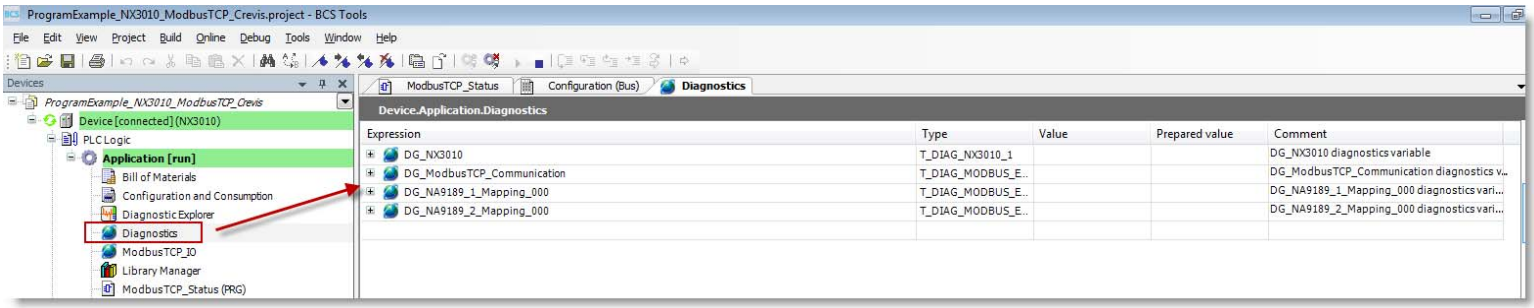
If a Modbus TCP slave is missing in the network this can be read by the diagnostic variable.



7 Communication interlock signal

For every module mounted in the Nexto system, BCS Tools creates diagnostic structures. These structures contains detailed information of the cpu, communication module, Profibus DP slave etc.

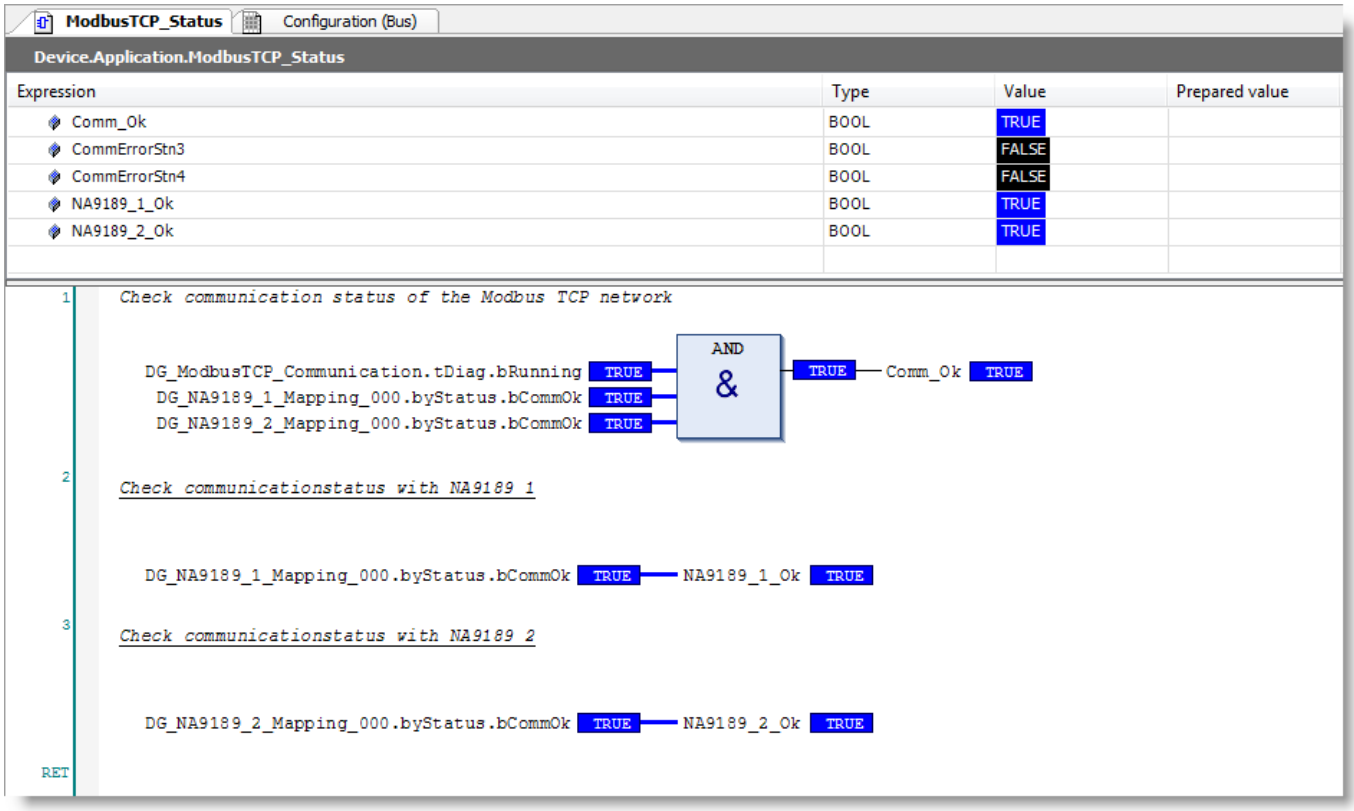
For more details about these structures refer to the User's manual of the specific module or the CPU User's manual.



7.1 Example of how to create a communication interlock signal

This is an example of how we can create a Comm_OK signal. When any of the configured slaves stops responding to the master this flag will be set to FALSE.

There is also flags available for independent communication control of each slave.



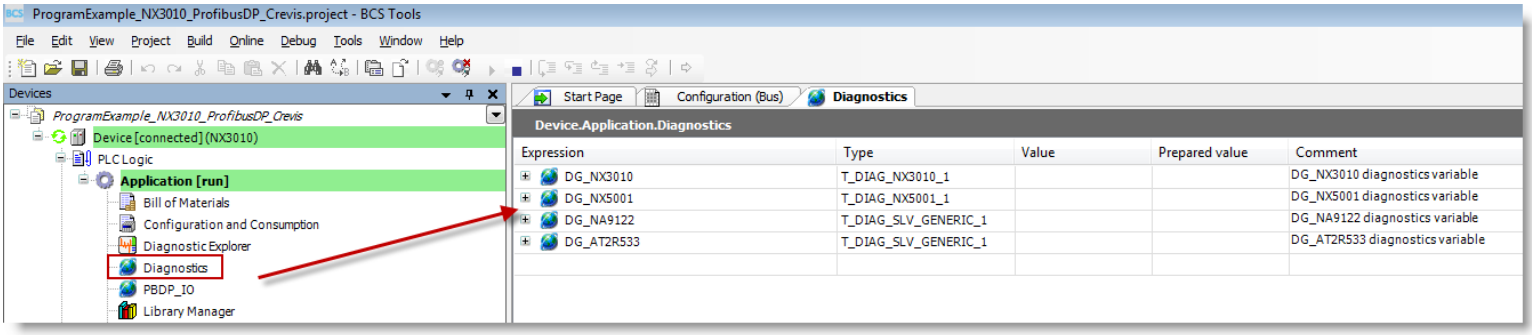
8 Troubleshooting

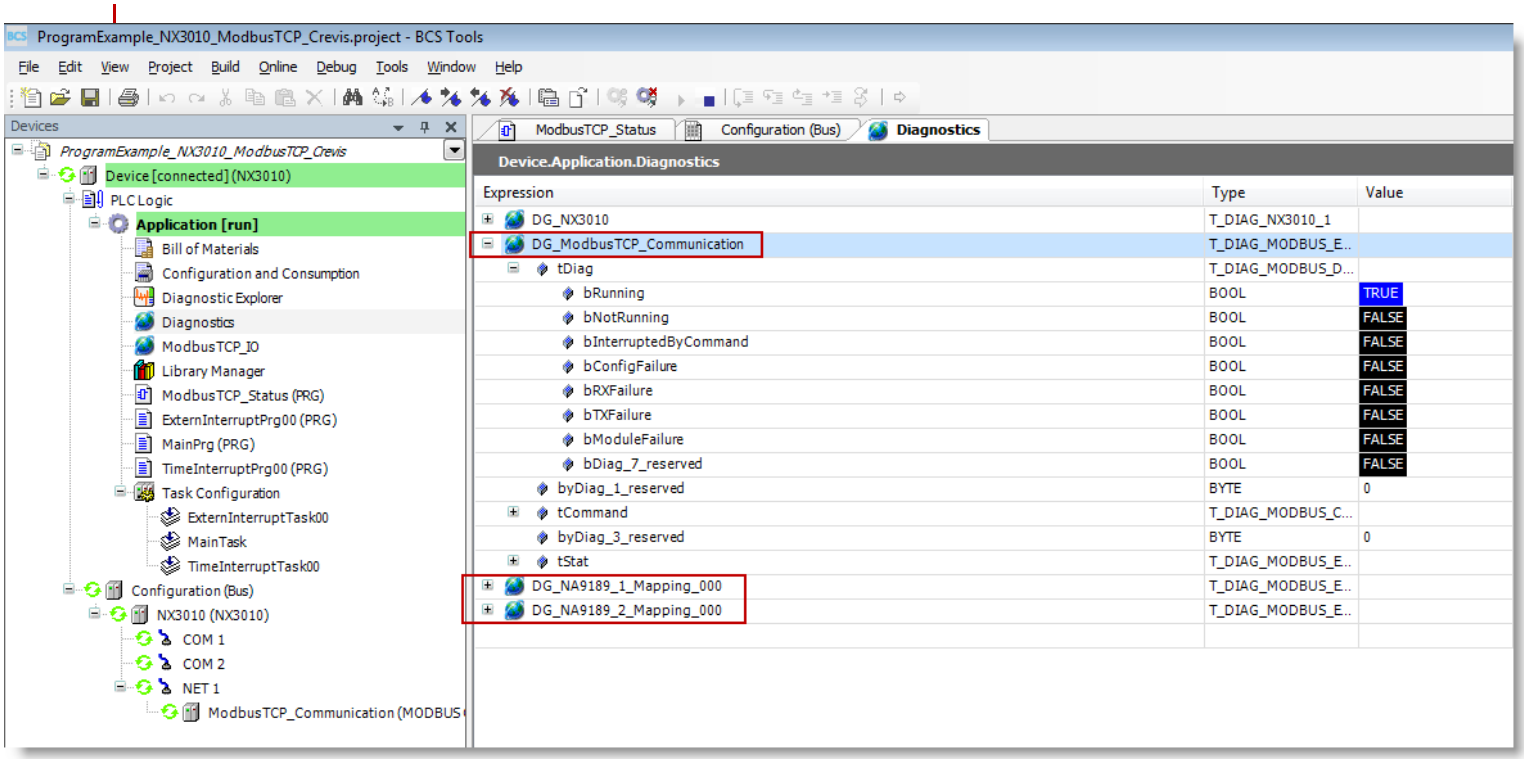
To troubleshoot and get diagnostics from the Nexto system there are several possibilities available.

- Diagnostics structures, created automaticly in BCS Tools.
- The OTD, One Touch Diagnostics function and LED:s.
- Diagnostics via the WWW page in the cpu.

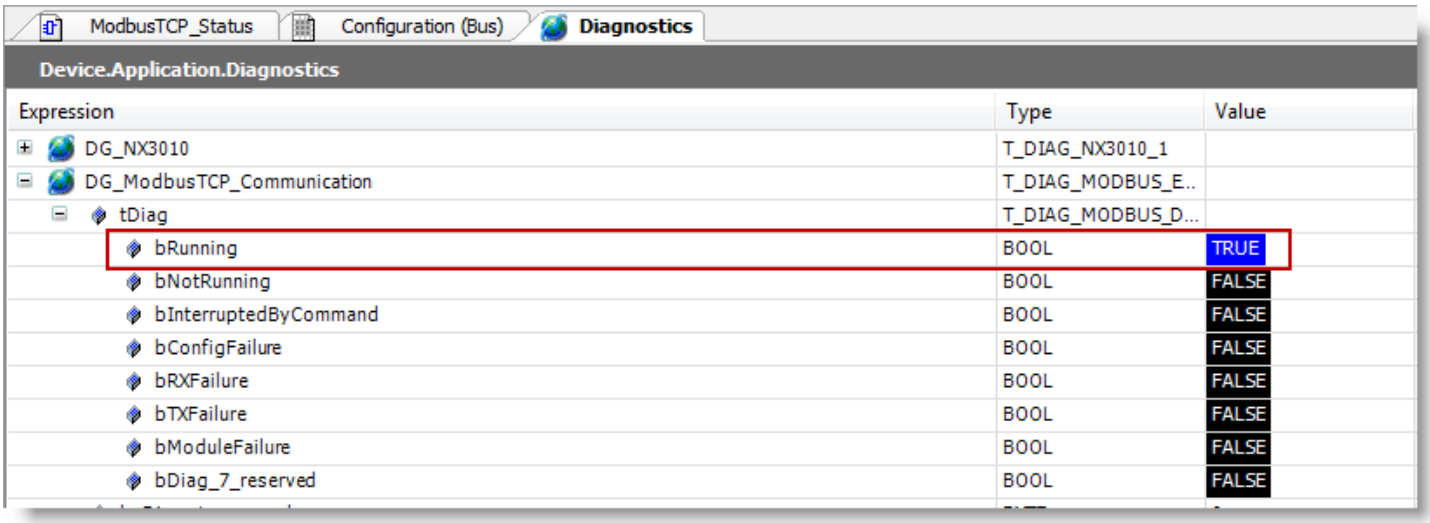
8.1 Diagnostic structures

When the Modbus TCP driver is inserted a data structure is created automaticly located in a list named Diagnostics. This list will be populated with diagnostics structures for all hardware inserted in BCS Tools.

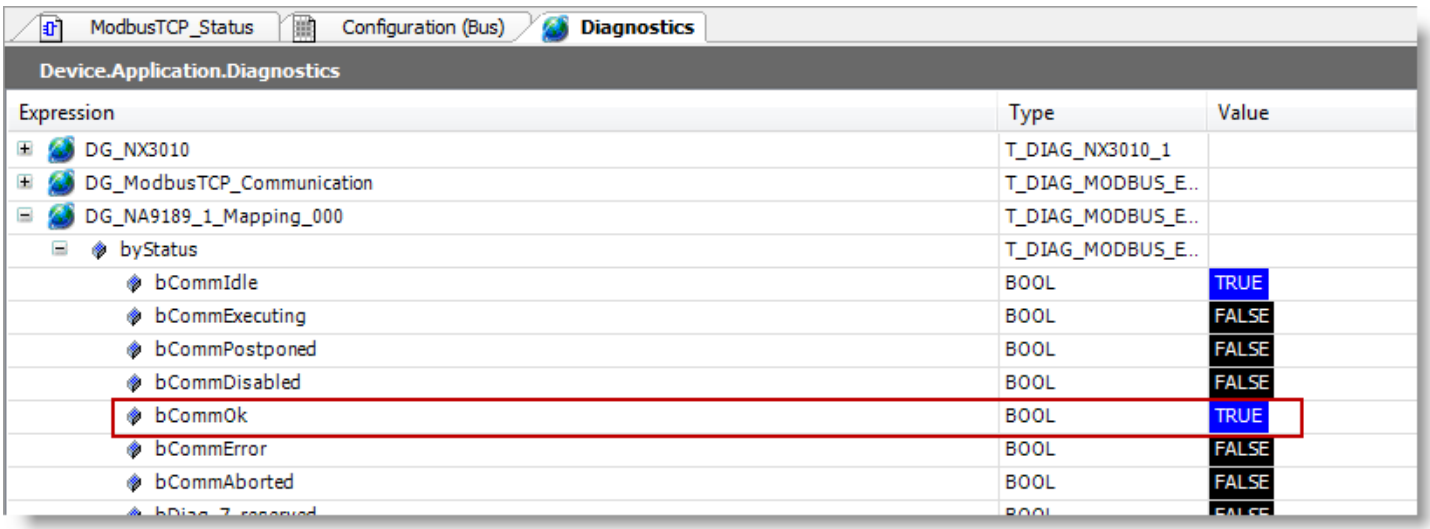




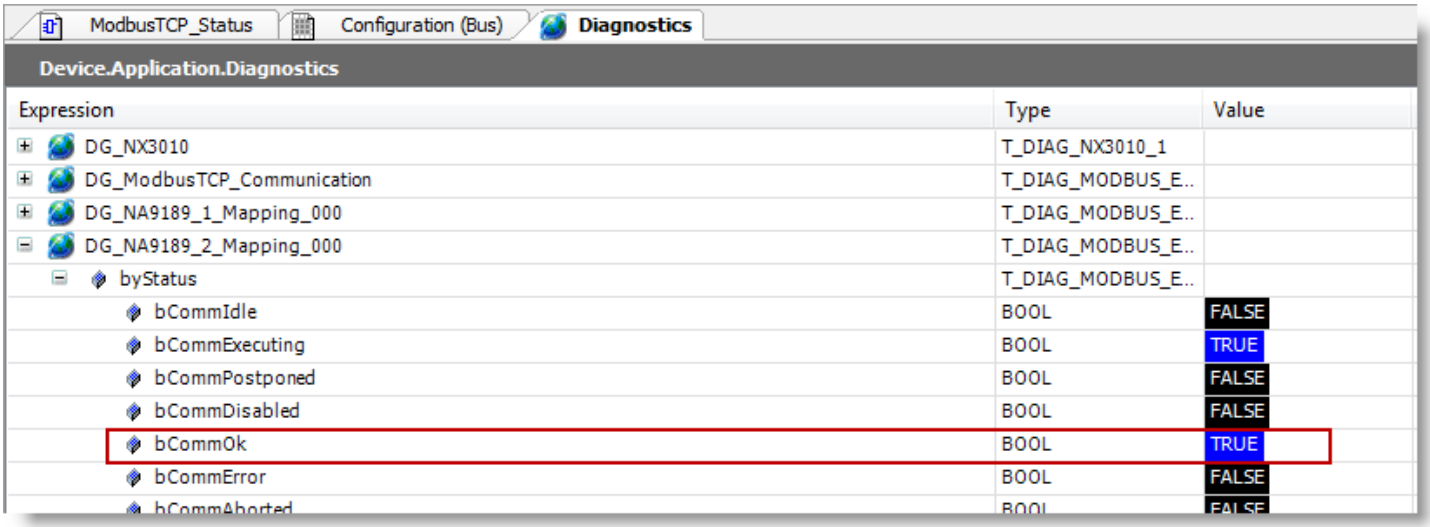
The diagnostics variables DG_ModbusTCP_Communication.bRunning is a variable indicating that the ModbusTCP driver is running with no errors.



The diagnostics variables DG_NA9189_1_Mapping_000.bCommOk is a variable indicating that the communication with the slave NA9189_1 is running with no errors.



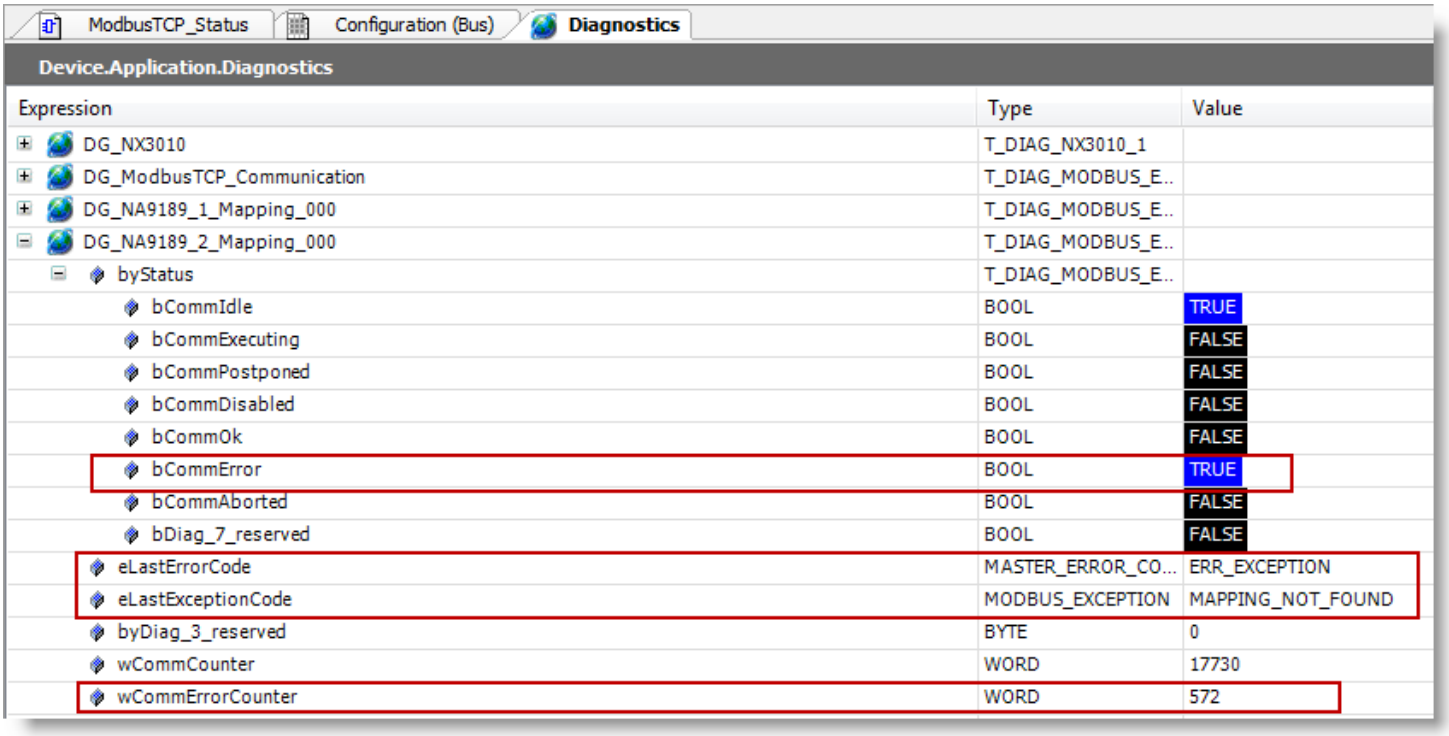
The diagnostics variables DG_NA9189_2_Mapping_000.bCommOk is a variable indicating that the communication with the slave NA9189_2 is running with no errors.



The following is an example when more data is accessed then available in NA9189_2.

The diagnostics variables DG_NA9189_2_Mapping_000.bCommError is TRUE, the DG_NA9189_2_Mapping_000.eLastErrorCode is reporting ERR_EXCEPTION and the DG_NA9189_2_Mapping_000.eLastExceptionCode is reporting MAPPING_NOT_FOUND.

When the error is present the DG_NA9189_2_Mapping_000.wCommErrorCounter will increment for every telegram sent to the slave.



Expression	Type	Value
DG_NX3010	T_DIAG_NX3010_1	
DG_ModbusTCP_Communication	T_DIAG_MODBUS_E..	
DG_NA9189_1_Mapping_000	T_DIAG_MODBUS_E..	
DG_NA9189_2_Mapping_000	T_DIAG_MODBUS_E..	
byStatus	T_DIAG_MODBUS_E..	
bCommIdle	BOOL	TRUE
bCommExecuting	BOOL	FALSE
bCommPostponed	BOOL	FALSE
bCommDisabled	BOOL	FALSE
bCommOk	BOOL	FALSE
bCommError	BOOL	TRUE
bCommAborted	BOOL	FALSE
bDiag_7_reserved	BOOL	FALSE
eLastErrorCode	MASTER_ERROR_CO...	ERR_EXCEPTION
eLastExceptionCode	MODBUS_EXCEPTION	MAPPING_NOT_FOUND
byDiag_3_reserved	BYTE	0
wCommCounter	WORD	17730
wCommErrorCounter	WORD	572

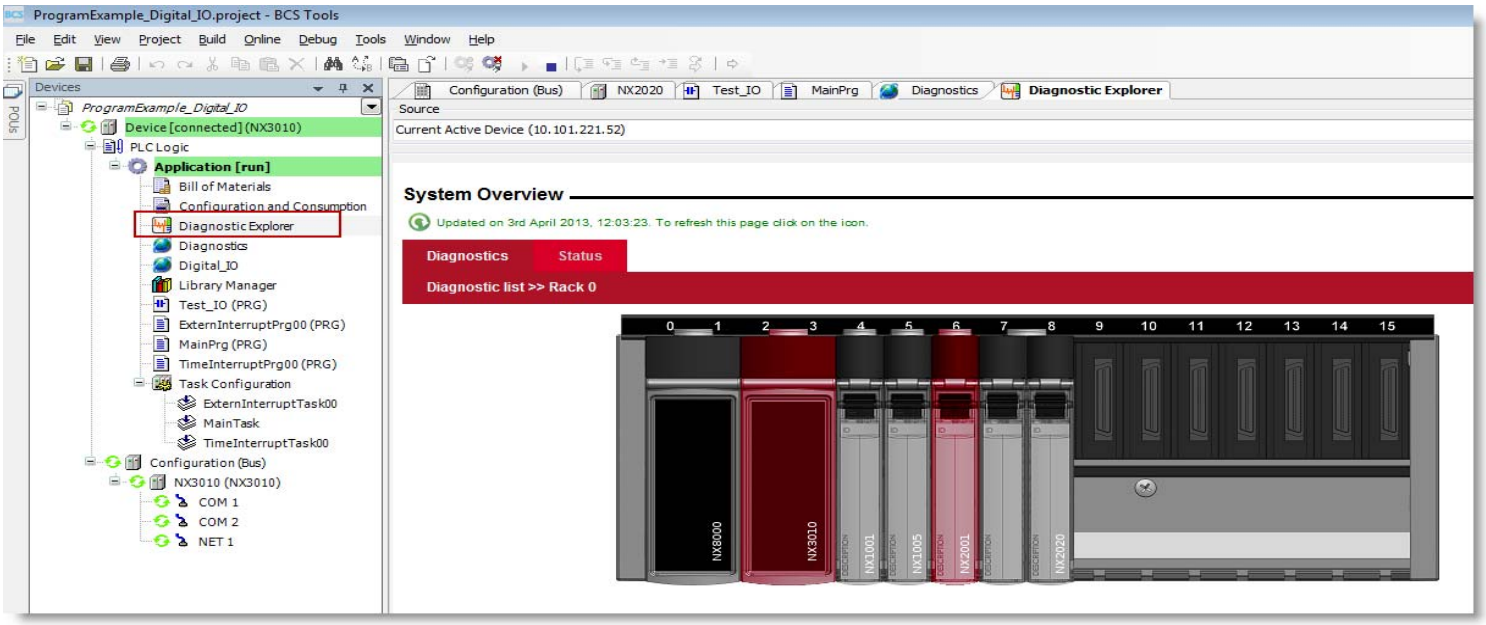
For more information about the structures refer to the User's manual of the cpu.

8.2 OTD and LED information

Note! When using the Modbus TCP driver the diagnostics is only available via the Diagnostic structures.

8.3 Diagnostics using the Diagnostic Explorer in BCS Tools

Note! When using the Modbus TCP driver the diagnostics is only available via the Diagnostic structures.



8.4 Diagnostics using the web browser

Note! When using the Modbus TCP driver the diagnostics is only available via the Diagnostic structures.